



TREBALL FI DE CARRERA

Títol

Estimació estocàstica de camps de transmissivitat condicionats a mesures de connectivitat (708-TES-EG-5698)

Autor/a

Ferran Agulló Gispert

Tutor/a

Daniel Fernàndez Garcia

Departament

Enginyeria del Terreny, Cartografia i Geofísica

Intensificació

Hidrologia Subterrània

Data

Juny 2012

RESUM

La modelització d'aqüífers ha estat durant molts anys, i és encara actualment, un dels objectes d'estudi principals de la hidrogeologia. La necessitat de poder predir el comportament de l'aigua i del transport de soluts en el subsòl ha originat diversos mètodes per a tal fi, essent l'estimació estocàstica part d'aquests. Tot i així segueixen existint carències en la caracterització de les propietats més importants dels aquífers com per exemple la connectivitat i el flux preferencial, i la influència que puguin tenir en altres paràmetres com ara la transmissivitat.

L'estimació estocàstica tradicional de mapes de transmissivitats es basa en la probabilitat que dos punts separats a certa distància tinguin valors similars. Malauradament, el flux d'aigües subterrànies i el moviment d'un solut en el subsòl depenen d'altres factors com ara la connectivitat hidràulica entre dos punts. Fet que fa que els mapes de transmissivitats obtinguts amb poca informació de la transmissivitat no puguin representar l'estructura de zones ben connectades.

L'objectiu de la tesina és estudiar l'estimació estocàstica com a metodologia per a fer mapes de transmissivitats condicionats a aquest tipus d'informació. Pel qual s'ha treballat amb FORTRAN a partir d'un programa elaborat per *Paolo Trinchero* (1) a la seva tesi doctoral "*Characterization of Spatial Heterogeneity in Groundwater Applications*". Finalment es compara els diversos resultats obtinguts amb el programa modificat, a partir dels mapes generats amb SGeMS.

Paraules Clau: Connectivitat, estimació estocàstica, Kriging, mesures dinàmiques.

ABSTRACT

The modelling of aquifers has been for many years and it is still one of hydrogeology main objects of study. The need to predict the water flow behaviour and solute transport in the subsurface has led to several methods for this purpose, being stochastic estimation methods part of them. However there are still gaps in the characterization of the most important aquifer properties such as connectivity and preferential flow, and the influence they may have on other parameters such as transmissivity.

The traditional stochastic estimation of transmissivity maps is based on the probability that two points separated at a certain distance have similar values. Unfortunately, the groundwater flow and the solute movement in the subsurface depend on other factors such as the hydraulic connectivity between two points. This makes transmissivity maps obtained with a little information on transmissivity not to represent at all the structure of connected areas.

The aim of this dissertation is to study the stochastic estimation as methodology to make transmissivity maps conditioned to such information. The work has been done using a FORTRAN program developed by Paolo Trinchero (1) in his doctoral thesis "Characterization of Spatial Heterogeneity in Groundwater Applications." Finally the different results obtained with the modified program are compared, using SgeMS to generate the maps.

Keywords: Connectivity, stochastic estimation, Kriging, dynamic metrics.

AGRAÏMENTS

En primer lloc voldria agrair la dedicació que en Daniel Fernández, tutor d'aquesta tesina, ens ha dedicat tant a mi com al treball realitzat, sempre amb un tracte pròxim i amable, que m'ha transmès la motivació necessària per tirar endavant la feina.

També voldria donar les gràcies a tots els amics i companys que m'han acompanyat durant tots aquests anys de carrera, i que han fet que el dia a dia fos més amè i divertit. Sense ells segurament no hauria arribat fins aquí. Voldria agrair especialment l'amistat del Rafa, per totes les aficions compartides, del Víctor, per ser el millor amic que una persona pot tenir, i l'Isaac, per la seva positivitat i alegria. Gràcies per donar-me una empenta quan calia.

Gràcies Alba, que tot i ser la última en arribar a la meua vida, ja farà dos anys que em fas la vida més senzilla, divertida i feliç. T'estimo "beluga", si algú forma part d'aquesta tesina ets tu, que tant m'has ajudat aquest últim any.

Voldria mostrar el meu profund agraïment a l'Isabel per criar-me i cuidar-me durant tant de temps. També voldria agrair especialment el suport incondicional de tota la meua família, la Maria Núria, les meves germanes Sara i Laura, i sobretot del meu pare, "el Senyor de la Llar", en Fede. Gràcies per aguantar-me tots aquests anys i donar-me suport en els estudis, i en totes les coses de la vida, passés el que passés, tinc un deute gegant amb tu.

Per últim m'agradaria "dedicar" aquest treball a la meua mare, l'Anna Maria, tot i fer anys que no estàs amb nosaltres, estic segur que t'hauria agradat viure aquests moments. Gràcies per donar-me la vida, que m'ha permès gaudir de tots aquests moments i persones meravelloses. Gràcies a tots de tot cor!

TAULA DE CONTINGUT

Resum.....	i
Abstract	iii
Agraïments	v
1. INTRODUCCIÓ.....	1
1.1. Introducció	1
1.2. Estat de l'art	1
1.2.1. Mètode Geoestadístic	1
1.2.2. Connectivitat	3
1.3. Objectius	5
1.4. Utilitat.....	6
2. METODOLOGIA ESTÀNDARD. MAPES DE TRANSMISSIVITAT AMB KRIGING.....	7
2.1. Introducció	7
2.2. Estimació estocàstica d'una sola variable	7
2.3. Interpolació mitjançant Kriging.....	12
2.4. Kriging puntual simple en el cas d'aplicació.....	14
3. MAPES DE TRANSMISSIVITAT CONDICIONATS A MESURES DINÀMIQUES DE CONNECTIVITAT	19
3.1. Introducció	19
3.2. Metodologia nova: Kriging amb mesures dinàmiques.....	19
3.3. Programa original	22
3.4. Obtenció de mapes de transmissivitat.....	24
4. EINES – PROGRAMARI.....	25

4.1. Introducció	25
4.2. SGeMS	25
4.3. Fortran.....	26
5. RESULTATS	29
5.1. Introducció	29
5.2. Anàlisi.....	29
5.2.1. <i>Origen de les dades</i>	29
5.2.2. <i>Aleatorietat de les dades</i>	33
5.2.3. <i>Mapes d'incertesa</i>	36
5.2.4. <i>Posició del pou</i>	38
5.2.5. <i>Anàlisi de sensibilitat numèrica</i>	41
6. CONCLUSIONS	45
7. BIBLIOGRAFIA.....	47
ANNEX A: CODI PRINCIPAL	
ANNEX B: ARXIU AMB PARÀMETRES D'ENTRADA	
ANNEX C: SUBROUTINES	
ANNEX D: SCRIPTS	

ÍNDIX DE FIGURES

Figura 1: Mapa conceptual del treball realitzat en aquesta tesina.....	6
Figura 2: Camp real. Logaritme natural de la transmissivitat ($Y=\ln T$).....	30
Figura 3: Mapes de Y generats a partir de 4 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	31
Figura 4: Mapes de Y generats a partir de 8 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	31
Figura 5: Mapes de Y generats a partir de 12 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	32
Figura 6: Mapes de Y generats a partir de 16 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	33
Figura 7: Mapes de Y generats a partir de 16 dades mesurades de Y. A) Equiespaiades, B) Aleatòries.	34
Figura 8: Mapes de Y generats a partir de 16 dades mesurades de τ . A) Equiespaiades, B) Aleatòries.	34
Figura 9: Mapes d'incertesa generats a partir de 4 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	36
Figura 10: Mapes d'incertesa generats a partir de 8 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	37
Figura 11: Mapes d'incertesa generats a partir de 16 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ	37
Figura 12: Camp real. Logaritme natural de la transmissivitat ($Y=\ln T$) amb posició dels nous pous d'estudi.....	38
Figura 13: Mapes de Y generats a partir de 4 dades mesurades de τ . Les formes geomètriques marquen les diferents posicions dels pous.	39
Figura 14: Mapes d'incertesa generats a partir de 4 dades mesurades de τ	40

Figura 15: Mapes de Y generats a partir de 8 dades mesurades de τ . A) Resolució = 1, B) Resolució = 0.1 42

Figura 16: Mapes de Y generats a partir de 16 dades mesurades de τ . A) Resolució = 1, B) Resolució = 0.1 42

1. INTRODUCCIÓ

1.1. Introducció

L'aigua ha sigut i és un element indispensable per a la vida, fet que ha dut l'home a estudiar el seu comportament, intentar comprendre els seus cicles i a controlar-la mitjançant obres de diverses envergadures.

Entre aquests estudis i accions s'hi engloba l'aprofitament dels recursos hídrics del subsòl, tant en fonts naturals com en pous, la construcció dels quals es remunta a l'antiguitat amb els primers assentaments humans, de manera que es pot dir que la hidrologia subterrània fa molts anys que suscita interès.

Avui en dia la hidrologia subterrània o hidrogeologia ha esdevingut una branca sòlida dins de la hidrologia, que estudia tant quantitativament com qualitativament l'aigua existent al subsòl. Existeixen diverses eines per realitzar aquests estudis i poder comprendre els mecanismes que governen els aqüífers, però es pot dir que l'eina més utilitzada actualment és la modelització. Els models permeten entendre, i fins i tot predir, el comportament de l'aigua en el subsòl, de manera que se'n pot treure el màxim profit i fins i tot controlar situacions nocives com ara la contaminació d'aqüífers.

Aquesta tesina pretén aportar un gra de sorra en l'estudi de la modelització d'aqüífers partint d'una part de la tesi doctoral de *Paolo Trinchero: "Characterization of Spatial Heterogeneity in Groundwater Application"* (1) , de la qual s'ha utilitzat com a base el programa per obtenir els resultats.

1.2. Estat de l'art

1.2.1. Mètode Geoestadístic

Trobar respostes i solucions als problemes hidrogeològics que ens presenten els sòls pot ser difícil i car, degut en gran part a la necessitat d'obtenir nombroses observacions de camp (mitjançant sondejos, assajos de traçador, mostreig del sòl, etc.) que descriguin la realitat el més acuradament possible. Tanmateix la complexitat d'aquests problemes requereix la presència d'un professional que analitzi les dades i ompli les carències de les mesures de

camp amb el seu coneixement i experiència del medi geològic, problemes de flux i transport, reactivitat, entre altres.

Tot això, pot no ser suficient, ja que tant l'obtenció de dades de camp com la comprensió del problema no són sempre complets, és a dir, no sempre se'n pot obtenir un resultat únic i precís. És en aquest punt on la geoestadística ofereix eines tals com els mètodes d'estimació estocàstica, els quals s'utilitzen per complementar tant les dades mesurades, com el coneixement dels professionals. Tenint en compte el grau d'incertesa d'aquestes, la contribució principal d'aquests mètodes és la de donar els millors valors estimats de punts desconeguts amb el seu error corresponent, és a dir fer una inferència d'una quantitat no mesurada, a partir de les dades mesurades.

Tal com el títol d'aquest document diu, el treball realitzat ha consistit en l'estimació d'un camp de transmissivitats. Això s'ha realitzat utilitzant mesures d'una variable en uns punts i estimant valors d'una altra variable a la resta de punts, donant un cert pes, importància, a les dades inicials segons uns criteris establerts, els quals s'explicaran detalladament junt amb el mètode general a l'apartat 2.

Segons *Montero et al.* (2) la geoestadística o teoria de variables regionalitzades va ser àmpliament utilitzada i popularitzada pel sector de la mineria durant la dècada dels setanta, i ara s'utilitza en varis camps de les ciències de la terra i enginyeria, sobretot en hidrogeologia. A més ha fet una gran aportació a l'estimació lineal de variables espacials, i ha popularitzat tant el variograma com la funció generalitzada de covariància.

Un dels problemes inicials més important de la geoestadística va ser la predicció de la riquesa d'un bloc a partir d'una mostra observada. Aquest procés va ser anomenant Kriging per Matheron (1963), en honor a D. G. Krige que ja als anys cinquanta va realitzar una bona discussió de com obtenir una primera predicció en la l'explotació d'or de manera selectiva a mines Sud-africanes. Tot i que la formulació del mètode no va ser elaborada per Krige i que autors com Wold (1938), Kolmogorow (1941) i Wiener (1949) van treballar en aquest tema, quan Matheron als anys seixanta va publicar "*Traité de Geostatistique Appliquée*" (3), i va formalitzar el mètode, va decidir posar-li el nom de Kriging per la contribució inicial de Krige. Així doncs s'utilitza el terme Kriging per descriure la predicció espacial òptima, que s'explicarà també al segon apartat.

1.2.2. Connectivitat

La connectivitat és el grau de connexió interna dels vèrtexs d'una xarxa territorial (4). Aquesta definició genèrica però, no s'adequa completament al significat que té en hidrologia subterrània. Aleshores, què és exactament la connectivitat? Tot i que la seva definició pugui no ser clara, es pot entendre el concepte com les cavitats limítrofes en un sòl o terreny, que formen un camí, per les quals l'aigua fluirà en cas que existeixi un gradient hidràulic.

Per entendre millor aquest concepte cal fer un lleuger repàs històric del que ha succeït en aquest camp fins a l'actualitat. A continuació, i basant-nos en l'article "*Connectivity metrics for subsurface flow and transport*" (5), fem una síntesis dels progressos que hi hagut en aquest camp en les darreres dècades i el seu estat en l'actualitat.

Aquests últims 50 anys la comprensió del funcionament d'aqüífers i la seva modelització ha anat evolucionant en tres fases principals. En la primera fase es considerava un conjunt de regions a les que se'ls atorgava unes propietats constants segons l'informació geològica de la que es disposava. En la segona fase la variabilitat a petita escala es va considerar com a factor clau i es va utilitzar la geoestadística per modelitzar la variabilitat espacial (6; 7; 8). Els paràmetres que descrivien el grau d'heterogeneïtat dels aquífers eren la variància del logaritme de la conductivitat hidràulica, i la seva longitud de correlació. D'aquesta manera es van obtenir un seguit de resultats importants (8; 9; 10; 11). Tanmateix diversos autors van senyalar que el model Multi-Gaussià era massa restrictiu i no era capaç de representar el catàleg complet de patrons de connectivitat que es poden trobar a la natura. El model Multi-Gaussià considera que les variables regionalitzades segueixen una funció de probabilitat estadística multinormal. En aquest punt comença la tercera fase, la situació actual, que a més dels paràmetres de la segona fase, considera la connectivitat com una propietat cabdal que influeix intensament en el flux subterrani i el transport de soluts en aquífers. Al mateix temps s'han desenvolupat un ampli ventall de models estocàstics per tal de representar estructures geològiques tals com canals, lents o deltes; amb l'objectiu de reproduir millor la connectivitat esperada d'aquestes estructures. Ha sigut en els darrers deu anys, però, quan hi hagut un auge en la publicació d'articles sobre el tema de la connectivitat en aquífers, molts d'ells fent referència a la problemàtica que no existeix una única definició matemàtica adoptada per tota la comunitat científica.

Encara que el concepte de connectivitat sigui ampli, es pot dir que es coincideix en la idea general del que representa, i per això seguidament s'explica com es pot quantificar aquest concepte. Es poden utilitzar indicadors estàtics i dinàmics.

- **Indicadors de connectivitat estàtics:** Només són funció de la distribució espacial dels paràmetres, com la litologia o la permeabilitat, i no consideren el procés físic que es dona a l'aquífer. És a dir, es tracta de mesures estadístiques en certs punts del terreny. Els mètodes estadístics més comuns per descriure la probabilitat de distribucions espacials són de dos punts, tals com la covariància o els variogrames, que cal notar, no són capaços de descriure la connectivitat. Aquest tipus de mètodes només donen la probabilitat que un punt tingui un cert valor depenent del valor d'un altre punt conegut; no hi ha cap influència de l'estructura o del procés que existeix entre ells (12; 13; 14; 15; 16; 17).

Així doncs pot semblar que els indicadors estàtics de connectivitat siguin de poca utilitat, però amb l'aparició de nous models d'heterogeneïtat que requereixen estadística més avançada (15; 18; 19), els indicadors estàtics estant esdevenint una eina important per tal de valorar la qualitat de les simulacions o per classificar-les (20; 21; 22; 23).

- **Indicadors de connectivitat dinàmics:** Els indicadors dinàmics són més diversos ja que depenen del tipus de procés. També representen els processos físics però depenen tant de la geometria com de paràmetres físics, com poden ser les condicions de contorn o l'estat del sistema. Per exemple en un medi no saturat, caldrà tenir en compte el grau de saturació, i de com aquesta afecta a la connectivitat. Això implica que els indicadors dinàmics també varien amb el temps. Tanmateix cal mencionar que són més fàcils de mesurar al camp que els indicadors estàtics.

Ambdós indicadors estan relacionats mitjançant un seguit de correlacions complexes (24; 17; 25; 26; 27; 28). Les quals no són objecte d'estudi d'aquesta tesina i que per tant no s'explicaran.

Per una altra banda també es poden classificar els indicadors depenent de si són globals o locals. Per una banda els globals, també coneguts com "*Geobody*" o "*Sand body connectivity*" (29), que es basen en la caracterització de l'estructura general del medi que proporciona la informació, la qual no té perquè estar relacionada amb un punt específic. A mode

d'aclariment, un exemple seria la probabilitat que dos punts presos a l'atzar del domini pertanyin a un mateix patró. Pel que fa als indicadors locals, també coneguts com "*Reservoir to well*" o "*Reservoir connectivity*" (29), estan relacionats amb un o diversos punts específics del medi. Un bon exemple seria voler conèixer el volum connectat a prop d'un pou, o saber si dos punts estan connectats mitjançant una assaig de traçadors. Aquests resultats només prenen un valor binari, és a dir, si està connectat, 1, si està desconnectat, 0. Amb un model estocàstic, també es poden obtenir resultats similars calculant la probabilitat que dos punts d'un model estiguin connectats (30). Es remarca que tot i ser anomenats indicadors locals perquè especifiquen un punt, no és una propietat local ja que el medi sencer hi pot estar involucrat.

Pel que fa la modelització, la connectivitat no acostuma a ser un paràmetre d'entrada explícit, sinó que ve implícit en altres propietats del medi generades per alguna tècnica de simulació. De fet, la relació entre la connectivitat i els paràmetres d'entrada de diversos models estocàstics ha estat analitzada (31; 32; 33), i poden ser usades com a guia per triar un model o un conjunt de paràmetres en cas de necessitat.

1.3. Objectius

L'objectiu d'aquesta tesina és l'avaluació d'una nova metodologia d'estimació de camps de transmissivitats mitjançant mesures dinàmiques. La seva comparació i anàlisi amb la metodologia estàndard permetrà discernir si es tracta d'un procés òptim per a l'estimació de mapes de transmissivitats, o no.

És a dir, partint d'unes poques dades inicials, extretes d'un camp real de transmissivitats per una banda, i el seu camp anàleg de temps d'arribada (connectivitat) per l'altra, es pretén trobar el camp de transmissivitats estimat que més s'ajusta a la realitat, o sigui al camp real de transmissivitats d'on s'ha fet l'extracció inicial. En el següent esquema queda reflectit el procés i l'objectiu amb més claredat:

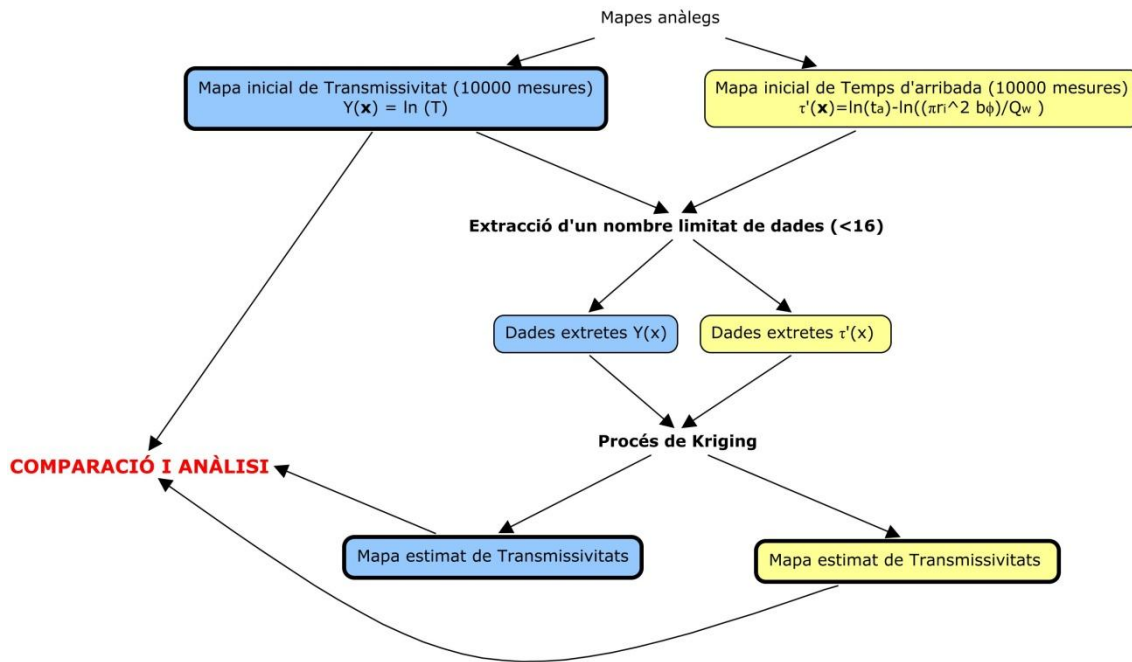


Figura 1: Mapa conceptual del treball realitzat en aquesta tesina

Com es pot veure a l'esquema, s'obtenen els camps estimats de transmissivitats mitjançant el mètode estadístic anomenat Kriging, el qual es descriurà amb detall en el següent capítol.

Així doncs, partint dels mapes estimats i comparant-los amb la realitat, l'objectiu és saber quin procés i quin tipus de dades inicials, són els que permeten obtenir un mapa estimat més similar al camp real de transmissivitats.

1.4. Utilitat

La utilitat d'aquest estudi és profunditzar en el coneixement de l'obtenció de dades per la modelització d'aqüífers i terrenys en general. Si s'aconseguís la certesa que a partir d'unes poques dades mesurades fos possible la recreació fidel de les propietats reals d'un aqüífer, no es requeririen tantes mesures de camp com assajos i sondejos, i per tant això suposaria un estalvi econòmic i de temps important. A més també suposaria un avantatge per les zones on no es pogués fer moltes mesures, tant per falta de temps com per accessibilitat.

Cal afegir que la finalitat última és comprovar la utilitat de mesures dinàmiques per a l'estimació de transmissivitats, el que reportaria nombrosos avantatges a l'hora de caracteritzar un aqüífer, tal com es veurà al llarg d'aquest document.

2. METODOLOGIA ESTÀNDARD. MAPES DE TRANSMISSIVITAT AMB KRIGING

2.1. Introducció

En aquest apartat s'explicarà els conceptes d'estadística bàsics que s'han requerit per realitzar l'estimació estocàstica duta a terme en aquesta tesina, tals com el variograma o la covariància, així com també el mètode de Kriging estàndard utilitzat per obtenir els camps de transmissivitat. La major part d'aquesta base teòrica s'ha obtingut del llibre *"Introduction to Geostatistics, Applications in Hydrogeology"* (34).

Així mateix, també es concretarà com s'han emprat aquests conceptes i mètode en el cas particular d'aquest treball.

2.2. Estimació estocàstica d'una sola variable

Es pot definir un procés estocàstic com un procés aleatori, és a dir que no segueix un comportament determinista. Els mètodes estocàstics parteixen de dades observades en certs punts d'un camp, a partir dels quals es simulem els valors de tots els altres punts desconeguts del mateix camp. Aquesta simulació serà diferent cada vegada que es repeteixi el procés estocàstic, és a dir aleatòria, raó per la qual aquests mètodes s'anomenen precisament estocàstics.

En el cas d'una zona concreta i delimitada de la qual es disposi d'un seguit de dades mesurades, com ara la transmissivitat o la porositat, i se'n vulgui estimar el seu valor en qualsevol altre punt del camp, es necessitaran unes expressions matemàtiques que descriguin com varia aquesta característica a l'espai. És a dir, es necessita un model de variabilitat espacial.

Cal tenir en compte però, que les variables utilitzades en hidrologia varien d'un punt a un altre d'una manera complexa i difícilment explicable. És per aquesta raó que les dades mesurades tenen gran importància, ja que en molts casos serviran com a guia per a desenvolupar un model empíric. Aquests models es podria dir que porten inclòs el concepte de probabilitat, ja que per descriure la variabilitat espacial s'utilitzen les mitjanes. És a dir, per reflectir que una característica fluctua al voltant d'un valor mig, el millor, és trobar una

equació que correlacioni les fluctuacions en dos punts del camp, depenent de la distància que els separa.

Així doncs, sigui quina sigui la variable (porositat, concentració química, etc), el seu valor depèn de la seva posició en el camp, que es pot representar com $z(\mathbf{x})$, on \mathbf{x} és l'índex de localització, que quedarà definit segons les dimensions de la zona d'estudi:

$$\mathbf{x} = x_1, \quad \text{or} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \text{or} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Aquesta funció $z(\mathbf{x})$ es coneix com a variable regionalitzada. Que formalment es pot definir com un procés estocàstic amb domini contingut en un espai euclidià n -dimensional \mathbb{R}^n , $\{z(\mathbf{x}): \mathbf{x} \in N \subset \mathbb{R}^n\}$. Si $n=2$, $z(\mathbf{x})$ pot associar-se a una variable mesurada en un punt \mathbf{x} del pla (35).

A efectes pràctics, el que es vol és estimar la variable $z(\mathbf{x})$ en una àrea, però normalment no s'obtindrà una solució única per l'escassetat d'informació inicial. Així, $z(\mathbf{x})$ és una d'un conjunt de possibilitats, $z(\mathbf{x};1)$, $z(\mathbf{x};2)$... Cada una d'aquestes possibilitats s'anomena realització, i el conjunt d'elles defineix totes les possibles solucions del problema d'estimació. Totes elles són diferents entre si, però comparteixen un patró i unes característiques generals. En el cas que es disposi d'un conjunt de realitzacions i s'hagi calculat la probabilitat que una d'elles sigui la incògnita actual, es pot escriure que la probabilitat que $z(\mathbf{x}) = z(\mathbf{x};i)$, per a qualsevol i , és P_i :

$$P_i = \Pr[z(\mathbf{x}) = z(\mathbf{x}; i)] \quad (2.1)$$

El conjunt de realitzacions amb les seves probabilitats assignades es coneix com a funció aleatòria, o procés espacial estocàstic.

Com que treballar amb el conjunt de realitzacions no és pràctic, s'acostuma a considerar que és més convenient utilitzar els moments estadístics. Així doncs interessa fer la mitjana de totes les realitzacions possibles:

$$E[z(\mathbf{x})] = \sum_i P_i z(\mathbf{x}; i) \quad (2.2)$$

La mitjana és el primer moment dels que s'acostumen a utilitzar en estimació lineal, i el segon és el moment anomenat covariància:

- Mitjana $m(\mathbf{x}) = E[z(\mathbf{x})]$

- Covariància $R(\mathbf{x}, \mathbf{x}') = E[(z(\mathbf{x}) - m(\mathbf{x}))(z(\mathbf{x}') - m(\mathbf{x}'))]$

Aquests moments permeten descriure el grau de suavitat de les realitzacions o l'escala de les fluctuacions, que són característiques importants dels conjunts de realitzacions, a més de ser més còmodes d'utilitzar que les probabilitats. Així doncs el model de l'estructura espacial consisteix en l'expressió matemàtica escollida per descriure la funció mitjana i la funció covariància. La seva importància radica en què caracteritza el conjunt de possibles solucions on es troba la funció desconeguda.

A partir dels conceptes anteriors es pot introduir l'anàlisi espacial estadístic, el qual consta de dues fases:

1. Anàlisi estructural
2. BLUE – “*Best linear unbiased estimation*”

La primera fase consisteix en la selecció i adequació de les expressions matemàtiques als moments de la variable regionalitzada, ja que la forma d'aquestes expressions és part del model que s'utilitzarà per fer les estimacions. Existeixen diverses expressions que es poden classificar en dos tipus:

- Les que serveixen per a propòsits generals, com per exemple el model intrínsec, que explicarem més endavant.
- Les que serveixen per a propòsits especials per a aplicacions específiques i que s'acostuma a personalitzar l'analista.

Per escollir el tipus d'expressió cal basar-se en les dades mesurades i en altra informació com pot ser l'experiència prèvia, localitzacions similars o la geologia i la hidrologia de la zona. L'analista ha de decidir quina informació és més rellevant, i escollir la funció més adequada mitjançant un procés iteratiu.

Per una altra banda a la segona fase es busquen valors estimats el més representatius i acurats possibles. Per aconseguir-ho, s'utilitza el model desenvolupat a la primera fase i les observacions específiques pròpies d'aquesta segona fase, les quals consisteixen en què totes les possibles solucions respectin les dades mesurades.

Com ja s'ha dit anteriorment existeixen diversos models amb diferents usos i nivells de complexitat. Aquesta tesina no pretén explicar-los tots a fons, però s'explicarà lleugerament el

model més bàsic, el model isòtrop intrínsec, per tal que posteriorment es pugui entendre la feina duta a terme.

Un dels models més simples és el que considera la mitjana (m) constant, i la covariància entres dos punts dependent, únicament, de la distància entre ells (h), tal com segueix:

$$E[z(\mathbf{x})] = m \quad (2.3)$$

$$E[(z(\mathbf{x}) - m)(z(\mathbf{x}') - m)] = R(h) \quad (2.4)$$

on

$$h = \|\mathbf{x} - \mathbf{x}'\| = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + (x_3 - x'_3)^2} \quad (2.5)$$

Les equacions (2.3) i (2.4) comprenen el que seria el model estacionari i qualsevol funció $z(\mathbf{x})$ que les satisfaci es considera que compleix amb estacionarietat de segon ordre. Al mateix temps també es tracta d'un model isotròpic ja que només depèn de la distància entre punts i no de la direcció del segment que els separa.

Per una altra banda, existeixen diverses funcions de covariància, però per a treballar amb estimacions, el més adequat són les funcions amb un espectre de potència continu (de l'anàlisi de Fourier) i una variància finita. Sense entrar a estudiar els anàlisis de Fourier, tot el que cal saber és que per una funció estacionària, el límit o *sill* $R(0)$ és finit i el valor de $R(h)$ tendeix a zero quan h excedeix el valor del rang.

A partir d'aquest punt, per interpolar amb aquest model, es necessita conèixer el valor de m i els paràmetres que puguin haver-hi a la funció covariància escollida, els quals possiblement es puguin extrapolar dels punts observats. En molts casos però, no es coneix el valor de la mitjana i s'ha d'inferir de les dades, per aquesta raó és convenient treballar amb el variograma.

$$\gamma(h) = \frac{1}{2} E[(z(\mathbf{x}) - z(\mathbf{x}'))^2] \quad (2.6)$$

Cal notar que altres documents, poden referir-se al variograma com $2\gamma(h)$, i a $\gamma(h)$ com a semivariograma. En aquesta tesina per raons pràctiques i d'ús, a $\gamma(h)$ se l'anomenarà variograma. Per altra banda, també cal destacar que es tracta d'un variograma teòric, una expressió matemàtica, i que per tant no s'ha de confondre amb el variograma experimental

obtingut de les dades. Així doncs per a una funció estacionària la relació entre el variograma i la covariància és la que segueix:

$$\begin{aligned}\gamma(h) &= \frac{1}{2}E[(z(\mathbf{x}) - z(\mathbf{x}'))^2] = \frac{1}{2}E[(z(\mathbf{x}) - m) - (z(\mathbf{x}') - m)]^2 \\ &= -E[(z(\mathbf{x}) - m) - (z(\mathbf{x}') - m)] + \frac{1}{2}E[(z(\mathbf{x}) - m)^2] \\ &\quad + \frac{1}{2}E[(z(\mathbf{x}') - m)^2] = -R(h) + R(0)\end{aligned}\tag{2.7}$$

El que es pot llegir com: el variograma és igual a el valor negatiu de la covariància més un valor constant, que resulta ser la variància.

$$\gamma(h) = -R(h) + R(0)\tag{2.8}$$

Utilitzant el variograma, considerem un model que té una mitjana no especificada però constant, i una diferència quadrada mitjana entre dos punts que depèn únicament de la distància entre ells. Les equacions (2.9) i (2.10) defineixen el model intrínsec isotròpic.

$$E[z(\mathbf{x}) - z(\mathbf{x}')] = 0\tag{2.9}$$

$$\frac{1}{2}E[(z(\mathbf{x}) - z(\mathbf{x}'))^2] = \gamma(h)\tag{2.10}$$

El model intrínsec i l'estacionari, tot i semblar iguals tenen algunes diferències importants. Els paràmetres que es necessiten per caracteritzar els models són diferents, i es necessita menys informació per fer-ho en el model intrínsec que en l'estacionari. Els dos models assumeixen la mitjana constant però el primer evita usar un valor numèric. A més, el model estacionari utilitza la funció covariància que requereix el variograma més la variància, ja que no pot ser reconstruït més enllà del rang només amb el variograma.

Així doncs, es pot dir que el model intrínsec és més general que l'estacionari. Cal notar, que tots els models estacionaris són intrínsecs ja que les equacions (2.9) i (2.10) segueixen les equacions (2.3) i (2.4) respectivament, mitjançant (2.8). Per una funció estacionària el variograma a grans distàncies és igual al del llindar o *sill*, $\gamma(h) = R(0) = \sigma^2$. Tanmateix, el més important és que no totes les funcions intrínseques són estacionàries. De fet com a norma pràctica, una funció intrínseca no és estacionària si el seu variograma tendeix a l'infinit quan h tendeix a l'infinit. Un exemple d'aquest tipus de funció és $\gamma(h) = h$.

Però, per què té importància el model intrínsec? Quin és el seu ús pràctic? La resposta és que es tracta d'un model simple que és útil pels següents casos:

1. Resumir informació incompleta i patrons en dades sorolloses.
2. Permet interpolar a partir de les observacions de $z(\mathbf{x})$, com veurem pròximament.

Existeixen restriccions matemàtiques a l'hora d'escollir la funció covariància o variograma. La raó d'aquestes restriccions és que l'expressió de la variància de la combinació lineal dels valors de $z(\mathbf{x})$, no pot ésser negativa. Aleshores, tant si s'utilitza la funció covariància com el variograma per a l'expressió, s'ha de procurar que no facin negativa, en cap cas, l'expressió (36; 37). Dins dels models estacionaris i intrínsecs n'existeixen de diversos tipus que no s'especificaran en aquest document però, que poden ser consultats a *Introduction to Geostatistics* (34). En el següent apartat, però, es parlarà del model adaptat en el treball realitzat.

2.3. Interpolació mitjançant Kriging

Una vegada assolits els conceptes anteriors, es pot definir el mètode utilitzat en aquest treball: Interpolació mitjançant Kriging. Suposant que tenim un conjunt de mesures, i assumint el model intrínsec isotròpic, el següent pas és escollir un variograma. Es pot utilitzar el variograma experimental però, per assegurar que és matemàticament acceptable, és millor escollir un variograma dels diferents models existents (34) i ajustar els seus paràmetres.

Abans de seguir amb les equacions de Kriging, s'exposa a continuació un resum del que es vol dur a terme. Triar un variograma, que és equivalent a triar un conjunt de solucions; s'assumeix que la funció buscada es troba dins d'aquest grup de funcions. Pot ésser que hi hagi funcions que no respectin les dades mesurades, per exemple, si s'ha mesurat z en un punt és lògic eliminar únicament les funcions que no tenen el valor correcte en aquest punt.

La idea és que no se segueix la pista a les realitzacions individualment, sinó que es procura calcular el valor mig i la variància del conjunt de funcions (les que han quedat després d'haver tingut en compte les dades mesurades). El valor mig esmentat és la mitjana condicional (o *best estimate*) i la variància és l'error quadrat mitjà d'estimació de la funció desconeguda $z(\mathbf{x})$. Per tal de facilitar l'anàlisi es tenen en compte només els *best estimates* que depenen linealment de les dades.

El Kriging suposa aplicar la metodologia de l'estimador no esbiaixat òptim (BLUE) a les funcions intrínseques. Les equacions involucrades en el mètode poden semblar complicades però la seva aplicació i implementació en programes és directe. Per explicar-la s'utilitzarà el

següent problema genèric. Donades n mesures de z a uns punts amb les coordenades espacials x_1, x_2, \dots, x_n , estimar el valor de z al punt x_0 . Un estimador no és més que una fórmula que utilitza dades per tal de trobar un valor estimat d'una quantitat desconeguda; en aquest cas només es focalitzarà l'atenció en estimadors que siguin combinacions lineals de les mesures, com per exemple:

$$\hat{z}_0 = \sum_{i=1}^n \lambda_i z(x_i) \quad (2.11)$$

Aleshores el problema es redueix a trobar un conjunt de coeficients $\lambda_1, \dots, \lambda_n$. La diferència entre el valor estimat \hat{z}_0 i el valor actual $z(x_0)$ és l'error d'estimació:

$$\hat{z}_0 - z(x_0) = \sum_{i=1}^n \lambda_i z(x_i) - z(x_0) \quad (2.12)$$

Evidentment cal buscar un bon estimador. Concretament es seleccionen uns coeficients que segueixin les següents premisses.

Sense biaix (*unbiasedness*). De mitjana l'error d'estimació ha de ser zero:

$$E[\hat{z}_0 - z(x_0)] = \sum_{i=1}^n \lambda_i m - m = \left(\sum_{i=1}^n \lambda_i - 1 \right) m = 0 \quad (2.13)$$

El valor de m , però, no és conegut. Aleshores per tal que l'estimador no sigui esbiaixat per qualsevol valor de m , es requereix que:

$$\sum_{i=1}^n \lambda_i = 1 \quad (2.14)$$

Imposant aquesta condició s'elimina el problema de la mitjana desconeguda.

Variància mínima. L'error quadrat mitjà d'estimació ha de ser mínim. Utilitzant la condició anterior es pot calcular en termes de variograma. Desenvolupant algebraicament s'obté:

$$E[(\hat{z}_0 - z(\mathbf{x}_0))^2] = - \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j \gamma(\|\mathbf{x}_i - \mathbf{x}_j\|) + 2 \sum_{i=0}^n \lambda_i \gamma(\|\mathbf{x}_i - \mathbf{x}_0\|) \quad (2.15)$$

L'expressió (2.15) es coneix com a funció objectiva perquè és la que es vol minimitzar mentre que l'expressió (2.14) es coneix com a restricció, ja que limita els valors que poden prendre els coeficients.

Seguidament cal trobar la solució del problema just plantejat. Es pot fer utilitzant multiplicadors de Lagrange, que és un mètode d'optimització estàndard (34), el qual consisteix en minimitzar el Lagrangià. Operant algebraicament s'arriba les condicions necessàries per minimitzar un sistema de Kriging lineal de $n + 1$ equacions i $n + 1$ incògnites:

$$- \sum_{j=1}^n \lambda_j \gamma(\|\mathbf{x}_i - \mathbf{x}_j\|) + v = -\gamma(\|\mathbf{x}_i - \mathbf{x}_0\|), i = 1, 2, \dots, n \quad (2.16)$$

$$\sum_{j=1}^n \lambda_j = 1 \quad (2.17)$$

On v és un multiplicador de Lagrange.

Normalment s'escriu el sistema del Kriging en notació matricial. Essent \mathbf{x} el vector de les incògnites, \mathbf{b} el vector del costat dret i \mathbf{A} la matriu de coeficients:

$$\mathbf{Ax} = \mathbf{b} \quad (2.18)$$

Resolent aquest sistema s'obté el valor dels coeficients lambda que permet a la fi trobar el valor \hat{z}_0 estimat.

2.4. Kriging puntual simple en el cas d'aplicació

En el treball realitzat en aquest document s'han estimat els valors de transmissivitat d'un camp a partir de mesures de transmissivitat ($T(\mathbf{x})$). La part del treball que s'explica a continuació constitueix només la primera part del treball que tracta d'estimar mapes de transmissivitat amb tècniques estàndard. Aquestes tècniques són comparades després amb els resultats de la nova eina generada que té en compte la connectivitat dinàmica. En el cas de les mesures de transmissivitat s'ha treballat amb el seu logaritme ja que permet imposar la no negativitat de la funció.

Pel que fa el mètode, s'ha utilitzat el Kriging simple ja que és més estable que el Kriging ordinari, degut a que aquest últim necessita més dades i podria ser més problemàtic a l'hora de fer els càlculs. S'ha pres aquesta decisió perquè facilita el càlcul i no afecta als resultats, ja que el que es vol estudiar no depèn del tipus de Kriging utilitzat. Per una altra banda, el Kriging simple es diferencia del Kriging ordinari, exposat a l'apartat anterior, per ser conegut el valor de la mitjana, de manera que es du a terme un canvi de variable que fa nul·la l'esperança de la nova variable, és a dir:

$$E[z(\mathbf{x})] = m \quad (2.19)$$

$$y(\mathbf{x}) = z(\mathbf{x}) - m \quad (2.20)$$

$$E[y(\mathbf{x})] = 0 \quad (2.21)$$

$$\hat{y} = \hat{y}(\mathbf{x}) = \sum_{i=1}^n \lambda_i y_i \quad (2.22)$$

Es calculen els coeficients λ_i imposant que la variància de l'error sigui mínima, i posteriorment operant, s'obtingran les equacions de Kriging.

$$\text{Var}[\hat{y} - y] = E[(\hat{y} - y)^2] \quad (2.23)$$

$$\begin{aligned} E[(\hat{y}_0 - y(\mathbf{x}_0))^2] &= E\left[\left(\sum_{i=0}^n \lambda_i y_i - y_0\right)^2\right] = E\left[\left(\sum_{i=0}^n \lambda_i y_i - y_0\right)\left(\sum_{j=0}^n \lambda_j y_j - y_0\right)\right] \\ &= E\left[\left(\sum_{i=0}^n \lambda_i y_i\right)\left(\sum_{j=0}^n \lambda_j y_j\right)\right] - 2E\left[\sum_{i=0}^n \lambda_i y_i y_0\right] + E[y^2] \\ &= \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j E[y_i y_j] - 2 \sum_{i=0}^n \lambda_i E[y_i y_0] + E[y^2] \end{aligned} \quad (2.24)$$

Com que $E[y(\mathbf{x})] = 0$ i la covariància és una funció que només depèn del desplaçament, es té que:

$$E[y_i y_j] = R[y(\mathbf{x}_i), y(\mathbf{x}_j)] = R[\mathbf{x}_i - \mathbf{x}_j] \quad (2.25)$$

$$E[y^2] = R(0) = \text{Var}[y] \quad (2.26)$$

Per tant l'expressió a minimitzar és:

$$E[(\hat{y}_0 - y(\mathbf{x}_0))^2] = \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j R[\mathbf{x}_i - \mathbf{x}_j] - 2 \sum_{i=1}^n \lambda_i R[\mathbf{x}_i - \mathbf{x}_0] + R[0] \quad (2.27)$$

Derivant l'expressió (2.27) i igualant a 0:

$$\frac{\delta}{\delta \lambda_i} E[(\hat{y}_0 - y(\mathbf{x}_0))^2] = 2 \sum_{j=1}^n \lambda_j R[\mathbf{x}_i - \mathbf{x}_j] - 2R[\mathbf{x}_i - \mathbf{x}_0] = 0, \quad i = 1, 2, \dots, n \quad (2.28)$$

Obtenint les equacions del Kriging:

$$\sum_{j=1}^n \lambda_j R[\mathbf{x}_i - \mathbf{x}_j] = R[\mathbf{x}_i - \mathbf{x}_0], \quad i = 1, 2, \dots, n \quad (2.29)$$

Que es pot escriure de forma matricial com:

$$\begin{pmatrix} c_0 & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad \text{on } c_{ij} = R(\mathbf{x}_i - \mathbf{x}_j) \quad i \quad c_i = R(\mathbf{x}_i - \mathbf{x}_0) \quad (2.30)$$

És un sistema de n equacions amb n incògnites $\lambda_1, \dots, \lambda_n$. El sistema té solució única ja que la matriu dels coeficients és la matriu d'autocovariància que sempre és definida positiva i, per tant invertible. Una vegada calculats els coeficients s'obté l'estimació \hat{y}_0 . A més substituint (2.29) en (2.27):

$$E[(\hat{y} - y)^2] = R(0) - \sum_{i=1}^n R(\mathbf{x}_i - \mathbf{x}_0) \quad (2.31)$$

L'expressió (2.31) indica que $\text{Var}[\hat{y} - y] < \text{Var}[y]$. Escrivint aquestes expressions en termes de la variable general s'obté:

$$\hat{z} = m + \sum_{i=1}^n \lambda_i (z_i - m) \quad (2.32)$$

$$\text{Var}[\hat{z} - z] = \text{Var}[z] - \sum_{i=1}^n \lambda_i R(\mathbf{x}_i - \mathbf{x}_0) \quad (2.33)$$

Amb el que s'obté el valor estimat en el punt desitjat i el seu error.

Finalment es vol emfatitzar que en aquest apartat s'ha volgut explicar la metodologia estàndard per l'estimació de punts mitjançant Kriging. És a dir l'estimació de mesures estàtiques a partir de altres mesures estàtiques. En el cas del treball, com ja s'ha dit, s'ha

utilitzat el Kriging simple estàndard, però també un Kriging condicionat a mesures dinàmiques, el qual s'explica en detall en el següent capítol, i que és el punt d'estudi principal d'aquesta tesina.

3. MAPES DE TRANSMISSIVITAT CONDICIONATS A MESURES DINÀMIQUES DE CONNECTIVITAT

3.1. Introducció

En aquests capítol s'exposa el procés utilitzat per obtenir els resultats que s'analitzaran en el capítol 5, així com el contrast amb el mètode original de partida i els canvis que s'ha hagut de realitzar. Per fer-ho s'explica el funcionament del programa original (pertanyent a la tesi doctoral de Paolo Trinchero (1) i desenvolupat pel tutor d'aquesta tesina, Daniel Fernández) en primera instància, per seguidament descriure les modificacions que s'hi ha dut a terme. Es posarà especial atenció en ressaltar les diferències amb la metodologia estàndard.

Els indicadors estàtics i dinàmics dels que s'ha parlat al capítol 2 tenen una rellevància especial en la modificació del codi, ja que el que s'estudia és l'obtenció de mapes de transmissivitat, mesura estàtica, a partir de mesures dinàmiques, en aquest cas de temps d'arribada. Aquest últim és una mesura que depèn del medi, i que per tant també depèn de la transmissivitat. Així doncs els canvis en el programa original estaran enfocats en aquesta perspectiva.

3.2. Metodologia nova: Kriging amb mesures dinàmiques

Com ja s'ha esmentat a la introducció d'aquest capítol, l'objectiu de modificar el programa era que permetés l'obtenció de valors estimats de transmissivitat en lloc de temps d'arribada. És a dir, es pretén que el programa estimi una mesura estàtica a partir de mesures dinàmiques, que són més fàcils de mesurar al camp i contenen informació més detallada sobre la seva estructura geològica, per veure si el resultat és millor que en el cas estàndard. En primer lloc cal analitzar el problema.

En aquesta part del treball s'estima el mapa de transmissivitats a partir de mesures de dues classes diferents: mesures de la mateixa variable i mesures de connectivitat dinàmica. Aquesta última es mesura mitjançant el temps d'arribada obtingut d'un assaig de traçador convergent. Aquest assaig consisteix en mesurar les concentracions obtingudes amb el temps en un pou de bombeig (a cabal Q constant) d'una substància química coneguda, injectada a certa distància d'aquest pou. El temps d'arribada t_a es defineix com el centre de massa d'aquesta corba,

$$t_a = \frac{\int_0^\infty t c dt}{\int_0^\infty c dt} \quad (3.1)$$

En el cas de la transmissivitat s'ha treballat amb el seu logaritme al igual que abans, i en el cas del temps d'arribada, aquests s'han hagut de transformar mitjançant l'expressió (3.3), l'obtenció de la qual està explicada al segon capítol de la tesis doctoral de Paolo Trinchero (1).

$$z(\mathbf{x}) = \ln(T(\mathbf{x})) \quad (3.2)$$

$$\tau'(\mathbf{x}) = \ln(t_a) - \ln\left(\frac{\pi r_i^2 b \phi}{Q_w}\right) \quad (3.3)$$

On r_i és la distància radial entre el pou i el punt d'injecció, Q_w és la velocitat de bombament per unitat d'espessor, ϕ la porositat, i b el gruix de l'aqüífer. Evidentment una vegada acabat tot el procés s'haurà de desfer la transformació, per tal de trobar el valor estimat.

El programa original partia de mesures de transmissivitat i temps d'arribada indistintament. En aquest estudi també es parteix d'aquest tipus de dades, pel que no va ser necessària cap tipus de modificació en aquest aspecte. Així doncs la modificació més important es troba en els valors de sortida (*output*) de programa.

Sabent això, quin és el canvi que s'ha de dur a terme en el procés? En primer lloc, i sabent que no cal realitzar canvis en la introducció de dades, s'analitza el càlcul dels pesos realitzat per el programa original, que es pot resumir en la següent operació matricial:

$$\begin{pmatrix} c_0 & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad \text{on } c_{ij} = R(\mathbf{x}_i - \mathbf{x}_j) \text{ i } c_i = R(\mathbf{x}_i - \mathbf{x}_0) \quad (3.4)$$

On c_{ij} és la covariància entre totes les dades, siguin de transmissivitat o de temps d'arribada, mentre que c_i és la covariància entre el valor a estimar i les dades, és a dir $R(\tau', Y_i)$ o $R(\tau', \tau'_i)$. Essent Y_i i τ'_i les dades d'entrada i τ' el valor que es vols estimar. Una vegada trobats els pesos (λ_i, μ_i) resolvent els sistema matricial, es resoldria la següent equació, mitjançant Kriging simple, per tal de trobar el valor estimat.

$$\hat{\tau} = \sum_{i=1}^n \lambda_i Y'_i + \sum_{i=1}^n \mu_i \tau'_i \quad (3.5)$$

$$Y' = Y - m \quad (3.6)$$

$$\hat{\tau} = \sum_{i=1}^n \lambda_i Y_i - m \sum_{i=1}^n \lambda_i + \sum_{i=1}^n \mu_i \tau'_i \quad (3.7)$$

Aquest és el procés seguit pel programa original per trobar els valors estimats de temps d'arribada. Per adaptar-ho al cas del treball, s'ha hagut de realitzar un canvi a la matriu dreta, on s'ha canviat τ' per Y' , de manera que els seus elements han quedat com $R(Y', Y'_i)$ o $R(Y', \tau_i)$. Tanmateix també existeix un canvi en l'equació (3.5), quedant de la següent manera:

$$\hat{Y}' = \sum_{i=1}^n \lambda_i Y'_i + \sum_{i=1}^n \mu_i \tau'_i \quad (3.8)$$

$$Y' = Y - m \quad (3.9)$$

$$\hat{Y} = m \left(1 - \sum_{i=1}^n \lambda_i \right) + \sum_{i=1}^n \lambda_i Y_i + \sum_{i=1}^n \mu_i \tau'_i \quad (3.10)$$

L'equació (3.10) és el pas final en la obtenció de valors estimats de transmissivitat, però no és l'últim canvi realitzat en el programa, ja que també es requereix la incertesa, que en el programa original, en el cas de Kriging simple, és la següent:

$$\text{Var}[\hat{\tau} - \tau'] = \text{Var}[\tau'] - \sum_{i=1}^n \mu_i R(\mathbf{x}_i - \mathbf{x}_0) \quad (3.11)$$

Havent realitzat ja el canvi en la matriu dreta de covariàncies, resta modificar el programa per tal que generi la variància de Y enlloc de la variància de τ , de manera que la incertesa quedaria de la següent manera:

$$\text{Var}[\hat{Y} - Y] = \text{Var}[\hat{Y}' - Y'] = \text{Var}[Y'] - \sum_{i=1}^n \lambda_i R(\mathbf{x}_i - \mathbf{x}_0) \quad (3.12)$$

Tot i haver treballat amb Kriging simple també s'ha modificat el programa per tal que pugui dur a terme Kriging ordinari, modificant el programa original com segueix:

Programa Original	Programa Modificat
$\hat{\tau}' = \sum_{i=1}^n \lambda_i Y_i + \sum_{i=1}^n \mu_i \tau'_i$ $\hat{\tau}' = \sum_{i=1}^n \lambda_i (Y_i + m) + \sum_{i=1}^n \mu_i \tau'_i$	$\hat{Y} = \sum_{i=1}^n \lambda_i Y_i + \sum_{i=1}^n \mu_i \tau'_i$ $\hat{Y} = \sum_{i=1}^n \lambda_i (Y_i - m) + \sum_{i=1}^n \mu_i \tau'_i$

Taula 1: Modificacions en les equacions de Kriging ordinari. No utilitzades en aquest treball.

Aquest quadre ens indica els canvis realitzat per tal de trobar el valor estimat dels punts desconeguts. Pel que fa la incertesa i el canvi de les operacions matricials, els canvis han sigut els mateixos que en el cas de Kriging simple, tot i les diferències entre mètodes.

Finalment el que s'obté és una nova metodologia, que es basa en estimar una mesura estàtica a partir de mesures dinàmiques. Aquest procés genera uns resultats, que una vegada transformats a mapes, permetran discernir si la metodologia és òptima i/o millor que la del mètode estàndard.

3.3. Programa original

La finalitat del programa original és l'obtenció d'un camp de temps d'arribada estimat de 10000 punts, partint d'unes poques dades mesurades de transmissivitat i temps d'arribada, mitjançant mètodes geoestadístics d'estimació condicionals, el Cokriging.

En aquest cas el mètode s'anomena Cokriging perquè s'estimen els valors a partir de dos tipus diferents de dades mesurades, la transmissivitat i el temps d'arribada. Això però, no suposa un canvi substancial en aquesta tesina i en el funcionament bàsic del mètode, ja que en l'estudi realitzat en cap cas es fa Cokriging, sinó Kriging, essent aquesta la raó per la qual no s'explica en detall el mètode (34). Així doncs s'utilitzarà la terminologia Kriging a partir d'aquest moment enlloc de Cokriging.

El programa pot utilitzar tant el Kriging ordinari com el simple. Creant al final un arxiu de sortida amb tots els valors de temps d'arribada estimats òptims, i la seva incertesa corresponent.

Inicialment es van generar mapes de connectivitat de transport punt a punt d'un aquífer sintètic donat. Tanmateix, es considera el camp de transmissivitat usat per *Sanchez-*

Vila et al. (38) i que es reproduïx al capítol 5. El domini computacional consisteix en una malla de 500x500 cel·les, cada una de 1x1 metres. El logaritme del camp de transmissivitats, $y(\mathbf{x})$, correspon a una realització individual incondicional d'una funció multiGaussiana aleatòria, caracteritzada per un variograma esfèric isotròpic amb una mitjana de zero, variància unitària, i una escala integral de 10 metres.

El programa necessita un arxiu amb les dades mesurades d'entrada on s'especifica la posició de cada punt en el pla, el seu valor i de quin tipus de variable es tracta, que seran els valors a partir dels quals s'estimaran els punts desconeguts. A més, també requereix l'especificació de les característiques de la malla amb la qual ha de treballar, així com la resolució.

Es va situar un pou d'extracció al centre del domini amb un flux d'extracció prescrit i constant, mentre que als límits exteriors es va fixar un descens zero. El domini es va ampliar artificialment per evitar possibles efectes del contorn, i els mapes obtinguts i amb els que es van treballar, són representatius d'un subdomini de 101x101 cel·les centrades al voltant del pou.

Per una altra banda es considera que el flux subterrani es mou bàsicament per l'efecte del bombeig del pou d'extracció, i per tant, no es té en compte el flux causat pel gradient natural de la zona. Per cada escenari, s'utilitza el programa per a reproduir patrons reals de connectivitat utilitzant l'estimació condicional. La funció covariància de $y(\mathbf{x})$ es modela utilitzant un variograma esfèric isotròpic amb una variància unitat i rang 20 metres.

Es pot dir que la finalitat d'aquest procés és l'obtenció de mapes de temps d'arribada, a partir de mesures estàtiques de transmissivitat i mesures dinàmiques de temps d'arribada. Els resultats assolits mostren clarament els camins preferencials del flux cap el pou de bombament (1), essent el final una metodologia útil per conèixer la connectivitat del terreny.

3.4. Obtenció de mapes de transmissivitat

Una vegada modificat el programa s'ha procedit a l'obtenció de mapes de transmissivitat pel seu posterior anàlisi. S'han generat quatre tipus de mapes diferents depenent del tipus de dades d'entrada i de la informació de sortida:

- Mapes de transmissivitat generats amb dades mesurades de transmissivitat.
- Mapes de transmissivitat generats amb dades mesurades de temps d'arribada.
- Mapes d'incertesa generats amb dades mesurades de transmissivitat.
- Mapes d'incertesa generats amb dades mesurades de temps d'arribada.

Tanmateix, per cada una de les categories anteriors, s'han generat diversos mapes amb diferents nombres de dades mesurades d'entrada, essent aquests: 4, 8, 10, 12, 14, 16, 20, 25, 36, 49, 64 i 144. Aquests s'analitzaran segons la necessitat al capítol 5.

4. EINES – PROGRAMARI

4.1. Introducció

Els mètodes d'estimació estadística mostrats al capítol 2 i 3, tot i ser pràctics, resultarien tediosos a l'hora d'estimar un gran nombre de punts, com és el cas d'aquest treball. Avui en dia, gràcies als ordinadors, és possible dur a terme els càlculs que requereixen els mètodes esmentats amb rapidesa i una relativa simplicitat.

En aquest capítol s'introduirà el programari utilitzat en aquest treball. També es vol mencionar l'ús de dos *scripts* per a l'extracció de dades del camp sintètic de transmissivitats i temps d'arribada, un creat pel mateix autor d'aquesta tesina i l'altra pel tutor, *Daniel Fernández* (Annex D).

4.2. SGeMS

SGeMS són les sigles per "*The Stanford Geostatistical Modeling Software*", que és un programa de codi obert per solucionar problemes relacionats amb variables espacials. Proporciona una interfície simple i còmode per a la visualització de variables espacials a la vegada que permet l'ús de diversos algoritmes.

Aquests algoritmes permeten aplicar Kriging sobre les dades introduïdes per l'usuari, però desafortunadament es tracta del mètode estàndard, i no s'ha pogut utilitzar en aquest treball, ja que es requeria un programa que pogués treballar amb mesures dinàmiques.

Tot i així, s'ha utilitzat SGeMS per visualitzar les dades resultants del programa que realitza el Kriging, obtenint els mapes de transmissivitat que han permès una posterior comparació i anàlisi de resultats, els quals es mostraran al capítol 5.

Aquest programa és d'accés lliure i pot ser descarregat de la pàgina <http://sgems.sourceforge.net/>.

4.3. Fortran

El codi original a partir del qual s'ha treballat es va elaborar amb Fortran, raó per la qual s'ha hagut d'adquirir des de zero el coneixement d'aquest llenguatge. El fet que l'esmentat codi estigués fet amb una barreja de llenguatge Fortran antic i modern, no ha facilitat el procés d'aprenentatge, però gràcies a les notacions del codi, a les indicacions del tutor i a la informació del llibre *"Fortran 90/95 for Scientists and Engineers"* (39), ha sigut possible desenvolupar el treball satisfactòriament.

Quan un ordinador executa un programa, executa una cadena molt simple d'operacions com càrrega, guardar, suma, resta i multiplicació. Cada una d'aquestes operacions tenen un únic patró, anomenat *"operation code"*, que les identifica. El conjunt d'aquests codis és el llenguatge que l'ordinador reconeix i executa. Desafortunadament aquest llenguatge pot ser complicat per a les persones, pel que es treballa amb llenguatge d'alt nivell, el qual uns programes especial anomenats compiladors, tradueixen al llenguatge *"operation code"* que entén la màquina.

Actualment existeixen diversos tipus de llenguatges d'alt nivell tals com C++, Basic, Java, entre d'altres, però entre tots ells es considera Fortran com el llenguatge preeminent per càlculs científics en general, ja que és el l'origen de la majoria de llenguatges computacionals científics. El nom Fortran es deriva de *"FORMula TRANslation"*, i la seva primera versió va ser desenvolupada entre els anys 1954-1957 per IBM per al seu ordinador *"Type 704"*.

Des de la primera versió, Fortran ha anat evolucionant incorporant noves característiques i possibilitat com les subrutines, un nou format de codi lliure, seccions de matrius, operacions matricials, tipus de dades parametritzades i interoperabilitat amb llenguatge C, entre moltes altres. Totes aquestes inclusions han anat modernitzat el Fortran fins arribar a la versió més actual Fortran 2008.

Per treballar en un llenguatge cal conèixer el seu alfabet, en el cas de Fortran aquest consisteix en 86 símbols. Pel que fa l'estructura d'un programa aquest està format per *"statements"* que bàsicament poden ser executables, si descriuen accions que fa el programa quan és executat, o no executables, si donen la informació necessària pels càlcul que ha de fer el programa. A partir de la mescla dels *"statements"* es forma el programa que generalment estan dividits en tres parts:

1. Declaració. Aquesta secció consisteix en un grup de “*statement*” no executables al començament del programa que defineix el nom del programa, el nombre i tipus de variables referents al programa.
2. Execució. Aquesta secció consisteix en un o més “*statements*” descrivint les accions que el programa ha de dur a terme.
3. Finalització. Aquesta secció consisteix en un o més “*statements*” que paren l’execució del programa i avisen al compilador que el programa ha arribat a la seva fi.

Cal tenir en compte que es poden inserir comentaris lliurement sempre i quan vagin precedits d’un símbol d’exclamació.

Les característiques d’aquest llenguatge de programació, el van fer adient per desenvolupar un programa que permetés aplicar la metodologia del Kriging, tal com s’ha explicat al capítol 3, i que a més no té les limitacions intrínseques d’un programa convencional.

5. RESULTATS

5.1. Introducció

En aquest apartat es presentaran els mapes resultants del programa i s'analitzaran en detall comparant-los entre ells. Es realitzaran una sèrie d'anàlisis comparatius dels mapes de transmissivitat segons l'origen i l'aleatorietat de les dades, un anàlisi de sensibilitat numèrica, un anàlisi segons la posició del pou d'extracció i, finalment, un anàlisi dels mapes d'incertesa.

5.2. Anàlisis

5.2.1. Origen de les dades

Les dades mesurades o observades no acostumen a ser molt abundants degut als elevats preus dels assajos per obtenir-los. Per tant es considera important discernir quin tipus de dades són les més adequades per tal de generar un mapa de valors estimats que reflecteixi lleialment la realitat. De fet, el cas més normal en hidrogeologia és tenir poca informació (menys de 15 valors).

Per tal tasca, es requereix un mapa de transmissivitats real d'on extreure les dades per fer el Kriging i poder-ne fer la comparació final. Com ja s'ha dit anteriorment s'utilitza un camp de transmissivitats creat per *Sanchez-Vila et al* (38) que es mostra a la figura 2.

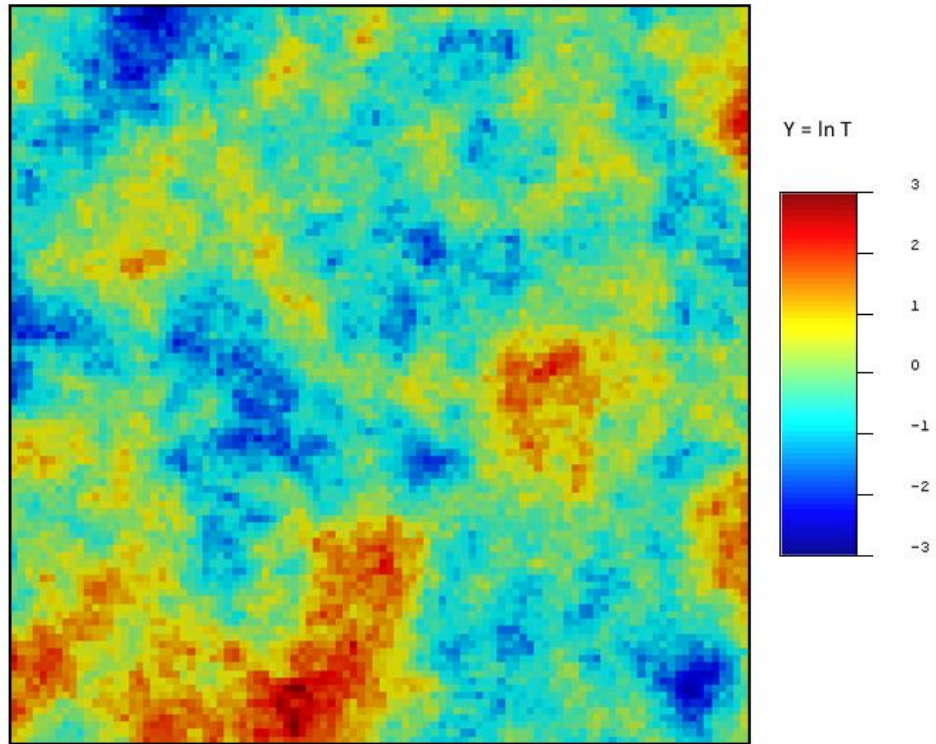


Figura 2: Camp real. Logaritme natural de la transmissivitat ($Y=\ln T$).

De tots els mapes realitzats es presenta només la selecció més representativa, i que es pot considerar real pel volum de dades mesurades. A més, en tots els mapes apareixen uns punts negres que marquen la disposició espacial de les dades i un punt verd que marca la posició del pou.

Per comparar la metodologia estàndard amb la proposada, es compararà amb cadascuna de les figures que es presenten a continuació el mapa de transmissivitat (o la seva incertesa) obtingut només utilitzant valors mesurats de transmissivitats en alguns punts, amb el mapa de transmissivitats obtingut només utilitzant valors de connectivitat dinàmica (τ') corresponents als mateixos punts de mesura.

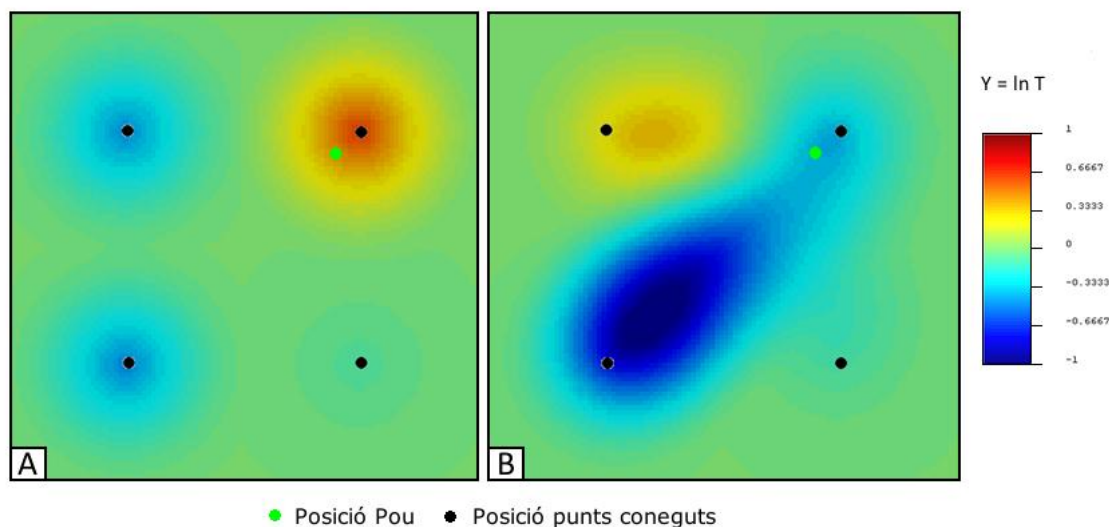


Figura 3: Mapes de Y generats a partir de 4 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ .

Els mapes de la figura 3 parteixen de poques dades mesurades i és un cas important per distingir el resultat segons les dades d'origen. A la figura 3A el mètode és capaç de captar valors estimats al voltant de les dades d'entrada, mentre que a la figura 3B es veu com el programa, amb el mateix nombre de dades, pot reflectir més acuradament les formes del camp de transmissivitats real, dibuixant la zona de baixa transmissivitat central.

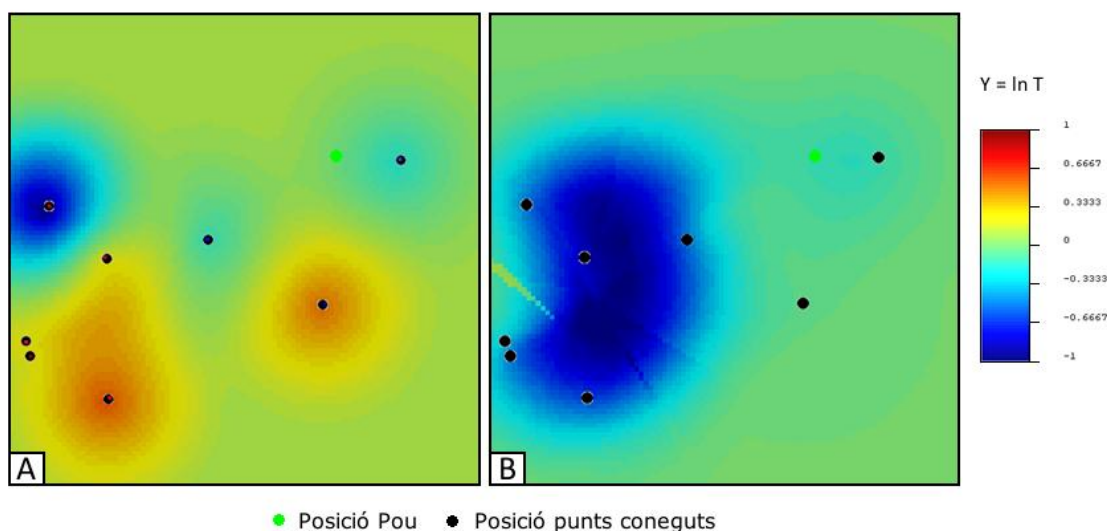


Figura 4: Mapes de Y generats a partir de 8 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ .

El cas de la figura 4 les dades van ser preses aleatòriament i la majoria d'elles es troben a la part esquerra del mapa. Aquest fet explica que a la part esquerra de la figura 4B hi hagi dibuixada amb una zona de baixa transmissivitat existent al camp real. Però, per què les zones d'alta transmissivitat no queden reflectides? Es creu que aquest fenomen es deu a que en el

recorregut entre els punts mesurats i el pou d'extracció no hi ha zones d'alta transmissivitat, i per tant el resultat del Kriging, que sempre està suavitzat, no ha pogut captar valors pic. Pel que fa el mapa 4A, succeeix com al mapa 3A, el mètode aconsegueix captar el valors estimats al voltant del punt, però depèn molt del valor mesurat del mateix punt.

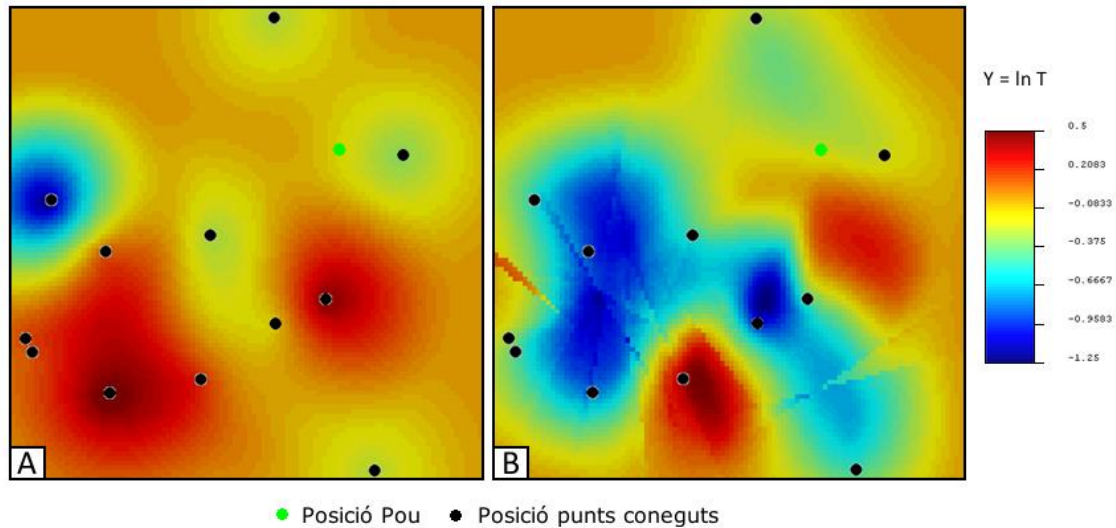


Figura 5: Mapes de Y generats a partir de 12 dades mesurades. A) Dades mesurades= Y , B) Dades mesurades= τ .

A la figura 5 les dades també han sigut extretes aleatòriament de la mateixa manera que el cas anterior però amb quatre mesures més. Es pot veure que el mapa 5A millora respecte els casos anteriors degut a l'augment de dades d'entrada, però malgrat tot encara no és una bona representació de la realitat. Pel que fa el mapa 5B s'observa amb claredat com el programa ha generat un esbós del mapa real (figura 2), dibuixant les zones amb transmissivitats més altes i més baixes. La diferència entre el mapa 5B i el 4B es creu que són les dades extremes afegides, les quals es troben en punts on la transmissivitat té valors pic, cosa que ajuda a definir el mapejat.

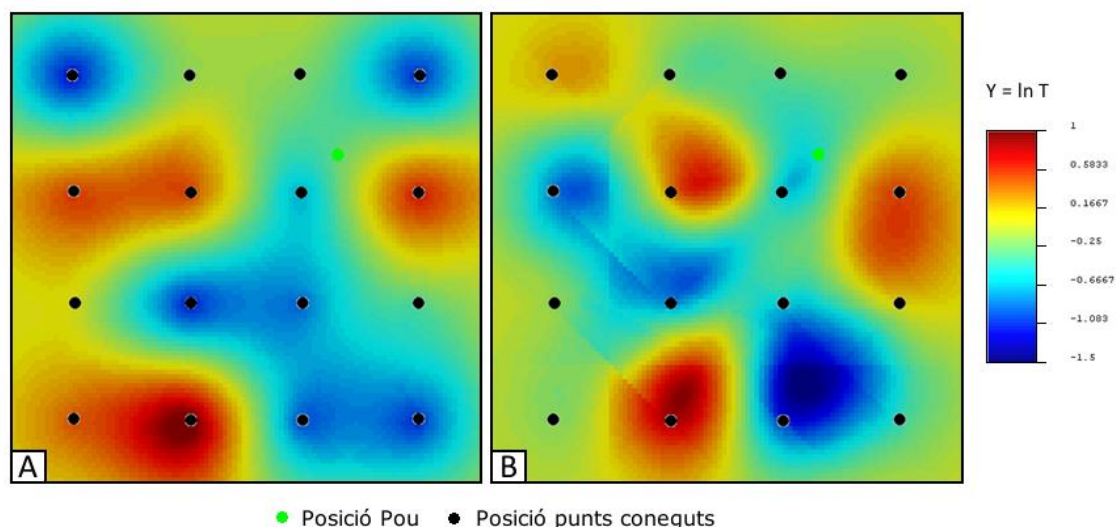


Figura 6: Mapes de Y generats a partir de 16 dades mesurades. A) Dades mesurades= Y , B) Dades mesurades= τ .

Finalment a la figura 6 es comparen els mapes resultants de setze mesures d'entrada equiespaiades a tota la zona. El que es pot observar és que per a un domini d'aquestes característiques, les setze mesures permeten que per primera vegada el mapa 6A tingui certa similitud amb el mapa real (figura 2). Pel que fa el mapa 6B, defineix molt millor les zones de pic de transmissivitat, sobretot al voltant del pou, podent-les reconèixer amb facilitat si es compara amb el mapa real de transmissivitats de referència. També cal dir que els contorns del mapa no queden del tot ben representats, fet que es deu, segurament, a que el flux va en direcció al pou.

5.2.2. Aleatorietat de les dades

A l'apartat anterior s'ha treballat amb mapes generats a partir de dades equiespaiades i aleatòriament repartides a l'espai. Sense haver comparat mapes amb el mateix nombre de dades, es pot intuir que els casos en què les dades es troben equiespaiades, el mapa generat és més similar al mapa real. Tot i així a continuació es comparen mapes amb 16 dades mesurades de partida, repartides de manera equiespaiada i aleatòria.

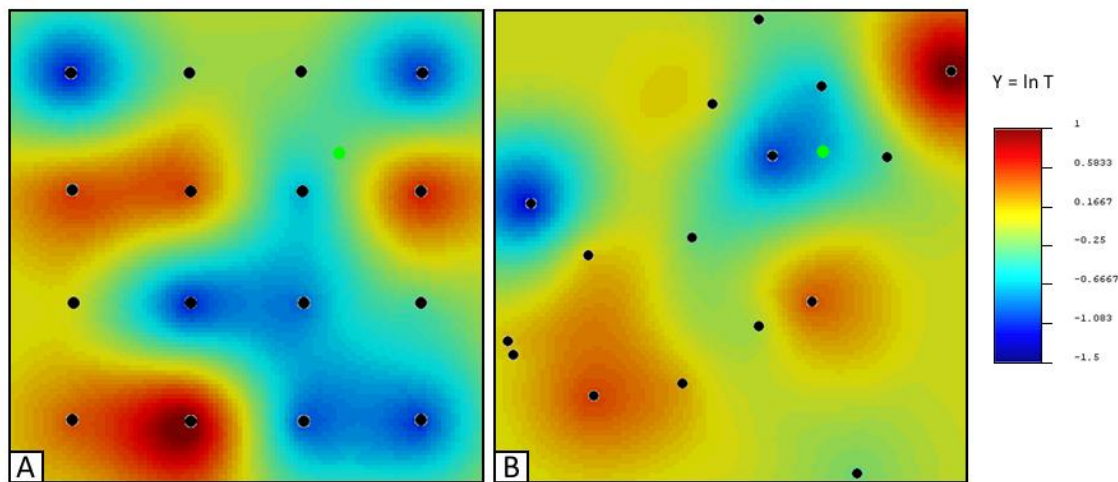


Figura 7: Mapes de Y generats a partir de 16 dades mesurades de Y . A) Equiespaiades, B) Aleatòries.

A la figura 7 s'observa com el mapa A, amb les mesures equiespaiades, representa millor els valors pic i, a més, al cobrir tot el domini, els valors en general són més pròxims als del cas real. Tot i així, al ser un mapa generat a partir de dades de transmissivitat, el resultat no és satisfactori del tot ja que es generen valors poc veraçs en certs punts, segurament degut a que els valors més pròxims tenen una influència molt alta. Pel que fa el mapa 7B, l'aleatorietat de les dades causa que en certes zones s'hi concentrin punts coneguts i en altres, no n'hi hagi o estiguin molt dispersos. Aquest fet causa que les zones amb més punts tinguin més ben definides les zones d'alta i baixa transmissivitat pel que fa la forma, essent la contrapartida, que les zones amb menys densitat de punts, quedin mal dibuixades i suavitzades pel que fa els valors de transmissivitat.

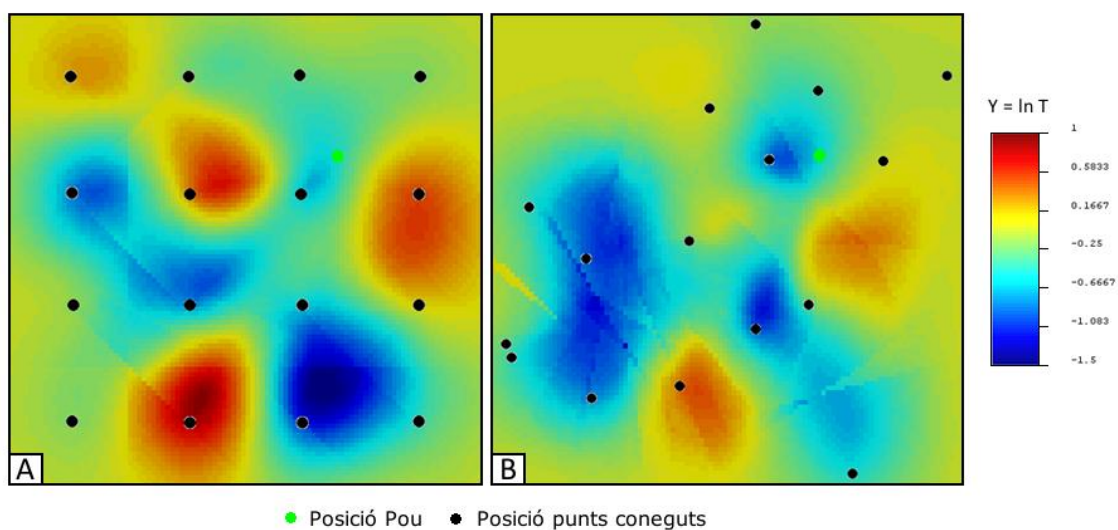


Figura 8: Mapes de Y generats a partir de 16 dades mesurades de τ . A) Equiespaiades, B) Aleatòries.

Els mapes de la figura 8, amb dades mesurades de temps d'arribada, succeeix aproximadament el mateix que a la figura 7. El mapa generat a partir de dades aleatòries té zones molt ben definides degut a l'alta concentració de punts (veure zona esquerra del mapa), mentre que les zones amb menys densitat de punts coneguts són més difusos. Tot i així aquestes últimes àrees queden més definides que al mapa 7B, com es pot veure si es compara la zona inferior dreta dels mapes 7B, 8B i la figura 2. Pel que fa el mapa 8A, les diferents zones de transmissivitat queden ben definides, essent la representació més similar al mapa de la figura 2. Cal fer notar, però, que en el cas de la figura 8 les zones d'alta i baixa transmissivitat tenen una millor definició que els fa més similars al mapa de referència (figura 2), tot i que, tal com ja s'ha dit a l'apartat 5.2.1, els contorns queden més difusos.

5.2.3. Mapes d'incertesa

Amb els mapes de transmissivitat també s'obtenen els mapes d'incertesa que ens indiquen quin grau de veracitat tenen els valors estimats. Aquests mapes resulten útils per comprovar com es calculen els punts desconeguts i saber quin és el grau de certesa de cada mapa. En primer lloc s'analitzen els mapes d'incertesa per 4 dades mesurades, ja que són els més clars.

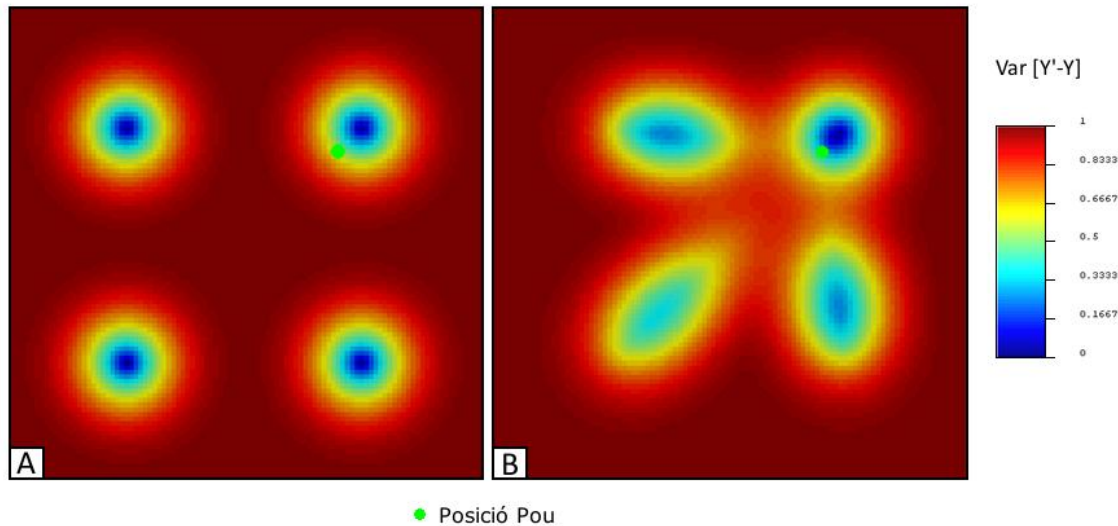


Figura 9: Mapes d'incertesa generats a partir de 4 dades mesurades. A) Dades mesurades=Y, B) Dades mesurades= τ .

A la figura 9A es pot veure com el grau d'incertesa és mínim en els punts mesurats i va augmentant radialment amb la distància. Això es deu al mateix funcionament del Kriging que estima els valors dels punts desconeguts segons la proximitat que tinguin amb el punt conegut. A la figura 9B succeeix aproximadament el mateix, la diferència radica en què les zones de certesa augmenten la seva àrea però perden intensitat, és a dir, tres dels punts estimats perden certesa màxima però guanyen punts de veracitat alta. Sembla que aquest augment es dona en direcció el pou, aquest fenomen s'estudiarà en detall en el següent apartat.

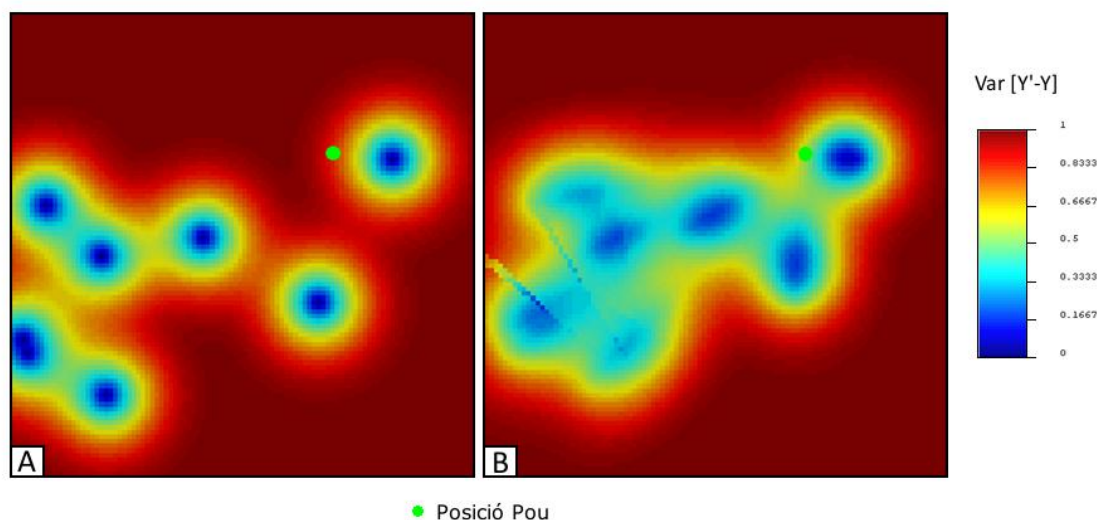


Figura 10: Mapes d'incertesa generats a partir de 8 dades mesurades. A) Dades mesurades= Y , B) Dades mesurades= τ .

Als mapes de la figura 10 es confirma el que ja s'ha vist a la figura 9. Aquest cas, però, al haver més punts coneguts i estar distribuïts aleatòriament, es pot confirmar que els punts de certesa augmenten al mapa 10B respecte al 10A, tot i que es perd qualitat en alguns punts del mapa 10A. També es torna donar l'estirament de les zones de certesa envers al pou.

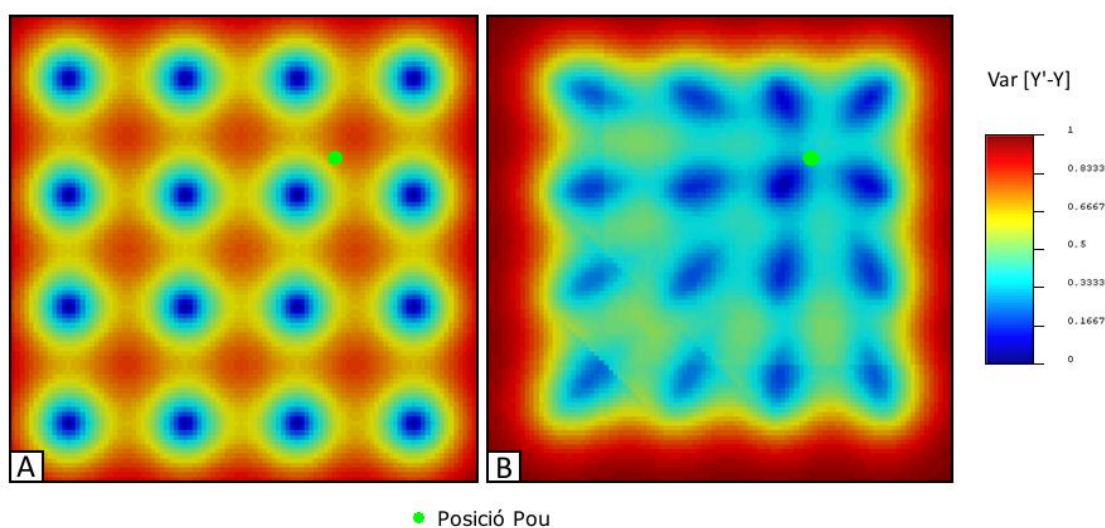


Figura 11: Mapes d'incertesa generats a partir de 16 dades mesurades. A) Dades mesurades= Y , B) Dades mesurades= τ .

Finalment s'analitzen els mapes d'incertesa per a 16 punts coneguts equidistribuïts en el domini (figura 11). Es confirma una altra vegada el que s'ha dit per la figura 9, i a més es pot veure clarament la raó per la qual els mapes de transmissivitats resultants de temps d'arribada tenen els contorn més mal definits que els resultants de dades de transmissivitat: La figura 11B mostra un contorn amb un grau molt alt, fins i tot màxim, d'incertesa.

5.2.4. Posició del pou

En tots els casos estudiats fins al moment el pou es troba en un punt, a mitja distància entre el contorn i el centre del domini, de transmissivitat mitja. Aquesta posició es deu a que no es desitjava una resposta extrema en les comparacions fetes fins el moment. Tenint en compte els mapes generats a partir de temps d'arribada, que són els més influenciats pel flux que bombeja el pou, s'ha fet un estudi del comportament del model segons la posició del pou.

La lògica fa creure que si el pou es col·loca en zones de baixa transmissivitat es tindrà més incertesa a l'hora d'estimar els punts desconeguts, ja que certs punts del domini poden notar poc o no notar la influència del pou. En cas de zones d'alta transmissivitat hauria de passar just el contrari.

La figura 12 és el mapa real de transmissivitats al qual s'ha marcat amb diferents formes geomètriques els nous pous amb els que s'ha fet corre el programa. S'han situat a zones d'alta i baixa transmissivitat, tant pròxims al centre com als contorns.

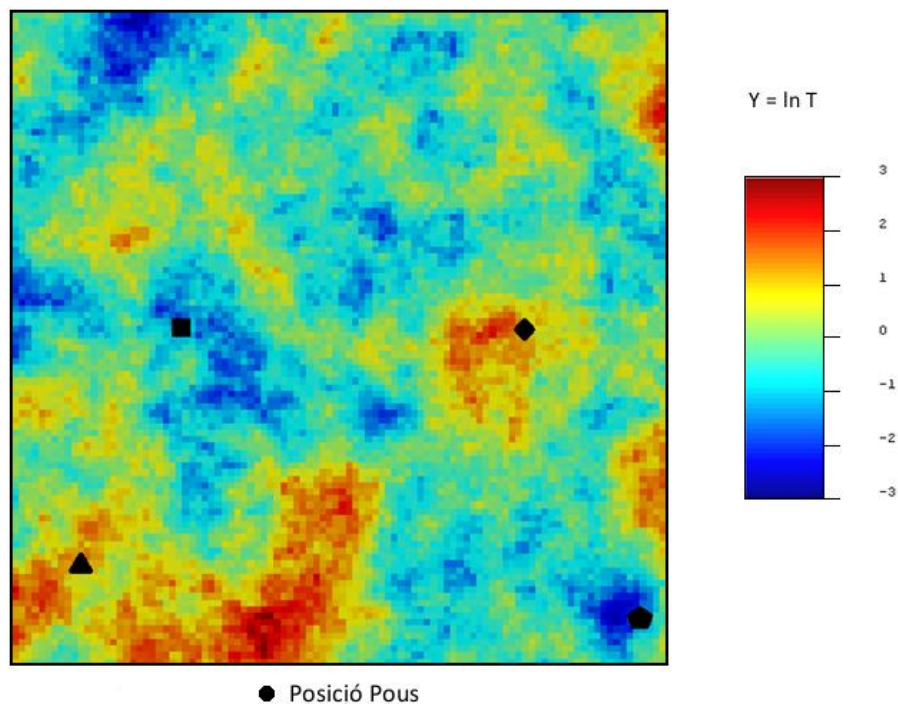


Figura 12: Camp real. Logaritme natural de la transmissivitat ($Y=\ln T$) amb posició dels nous pous d'estudi.

A continuació es mostren els quatre mapes de transmissivitats resultants per cada localització del pou. S'han escollit els mapes generats a partir de quatre dades per ser els més

simples, i els que permeten una millor lectura del que es vol analitzar. La posició de les dades és la mateixa que la que mostra la figura 3.

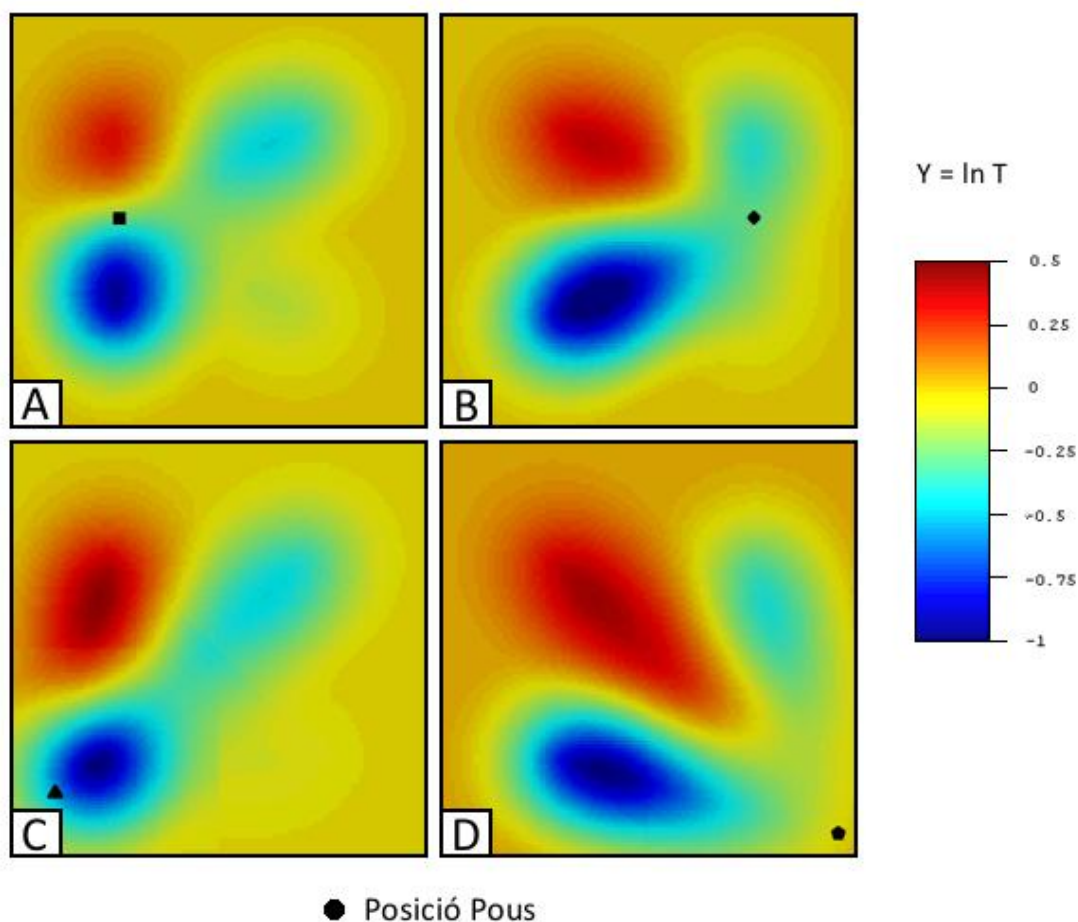


Figura 13: Mapes de Y generats a partir de 4 dades mesurades de τ . Les formes geomètriques marquen les diferents posicions dels pous.

La figura 13 mostra clarament que la posició del pou influeix en el mapa estimat resultant. El més destacat però és que les suposicions lògiques fetes abans de fer córrer el programa no semblen complir-se. Si s'observa la figura 13D, el pou està situat en una zona de baixa transmissivitat molt pròxima al contorn, i tot i així els punts coneguts el noten i estimen els valors dels punts desconeguts en direcció al pou preferentment, sense tenir en compte la transmissivitat del punt on es troba. A la figura 13C el pou està situat en una zona d'alta transmissivitat però, comparant-lo amb el mapa 13D, sembla que es trobi en una zona de més baixa transmissivitat. Pel que fa els pous de les figures 13A i 13B, es troben en posicions simètriques pel que fa el domini, però en zones de baixa i alta transmissivitat respectivament. Amb aquestes figures, però, és difícil saber si la situació del pou en un punt d'alta o baixa transmissivitat té més influència, que el fet de canviar el pou de posició.

Quina és la causa d'aquest comportament? Els mapes de la figura 13 no permeten fer cap suposició sòlida, i a més no donen massa informació a part que la posició del pou canvia els mapes resultants de mesures de temps d'arribada. Per aquesta raó s'ha decidit analitzar els mapes d'incertesa de cada un dels mapes de la figura 13:

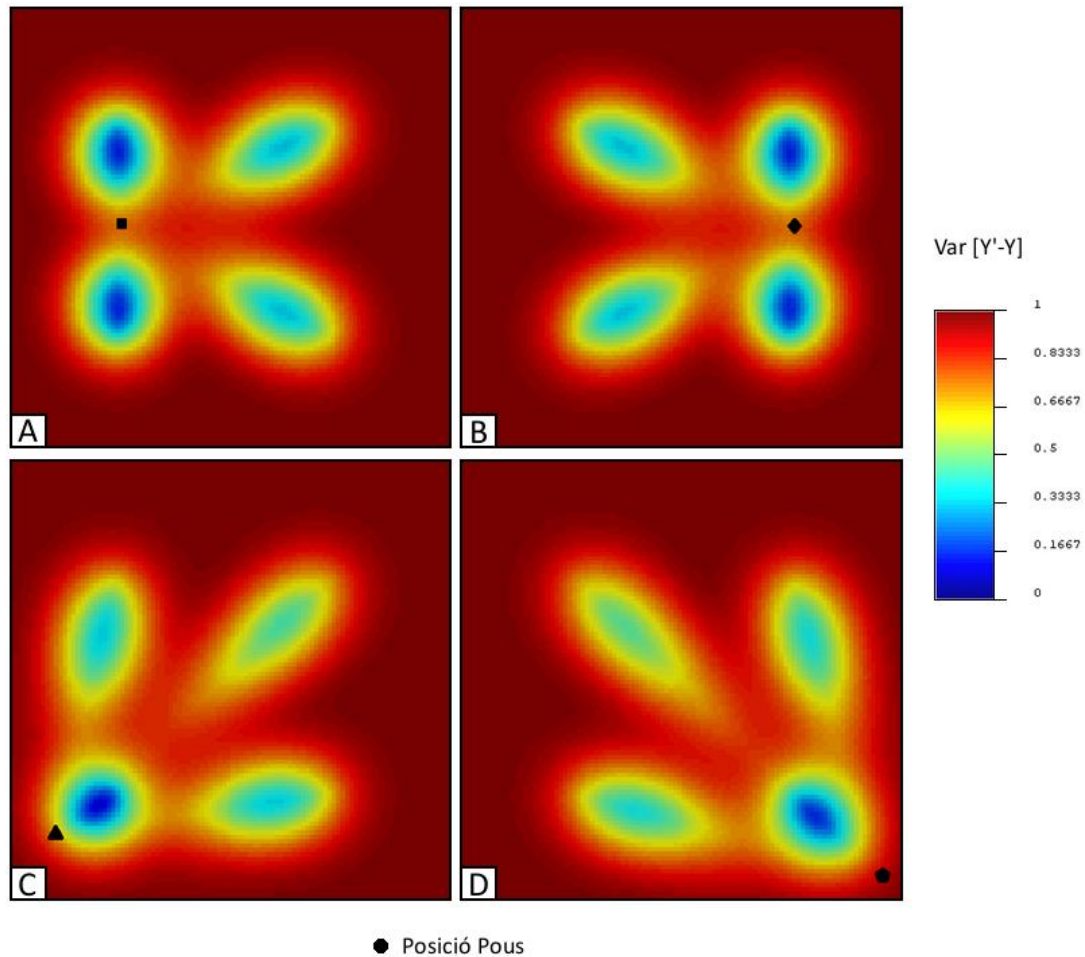


Figura 14: Mapes d'incertesa generats a partir de 4 dades mesurades de τ .

La figura 14 ajuda a discernir el que realment passa al canviar el pou de posició. Es pot veure clarament l'efecte de bombeig del pou sobre els valors estimats per cada punt, de manera que les zones més veraces queden estirades en direcció al pou. El que també es pot confirmar a partir d'aquesta figura és que la localització del pou en una zona d'alta o baixa transmissivitat no té una influència significativa en l'estimació de la resta de punts.

Aquest fet es podria deure a que el cabal de bombeig és molt alt de manera que no importa en quin tipus de terreny es trobi el pou, ja que capta aigua de tot el domini. Un altre factor a tenir en compte, que es pot intuir a la figura 13 i que a la figura 14 ja es veu clarament, és que el programa estima amb més certesa els punts al voltant de les línies rectes entre els

punts mesurats i el pou. Això es deu a que el Kriging és un mètode d'estimació que treballa amb la distància entre punts, i aquesta és en línia recta. En el cas de treballar amb dades procedents de temps d'arribada aquest fet és poc real, ja que l'aigua seguirà el camí més permeable, que no té perquè ser la línia recta, entre els dos punts.

5.2.5. Anàlisi de sensibilitat numèrica

Al llarg dels apartats anteriors s'han comparat diversos mapes de transmissivitats, entre ells els generats a partir de mesures de temps d'arribada. Una característica comuna en d'aquests últims, sempre i quan contenen més de quatre punts coneguts, és que tenen anomalies, les quals es presenten com esquerdes o talls de la continuïtat en la transmissivitat (figures 4B, 5B, 6B, 8A, 8B i 10B). Aquestes anomalies no es donen als mapes generats a partir de mesures de transmissivitat, i es caracteritzen per anar augmentant i ser més nombroses a mida que augmenta el nombre de mesures dinàmiques de temps d'arribada.

Com s'ha explicat anteriorment, el procés d'obtenció dels mapes de transmissivitat a partir de les mesures dinàmiques és més complex i car computacionalment, ja que no es tracta d'una relació directe de Kriging, sinó que les mesures dinàmiques depenen de les estàtiques, mitjançant una expressió integral, pel que cal trobar un seguit de relacions (1) que permetin el càlcul.

Totes aquestes observacions donen a entendre que aquestes anomalies són causades per un problema numèric. Així doncs s'ha dut a terme un anàlisi de sensibilitat numèrica per confirmar-ho i veure si es trobava una solució. S'han calibrat dos paràmetres per comprovar si els resultats milloraven, el primer d'ells, el radi de cerca de punts, no ha generat cap canvi apreciable en els mapes generats, pel que no es considera oportú esplaïar-s'hi. Pel que fa el segon dels paràmetres, la resolució de la integració lineal mitjançant una quadratura trapezoidal, si que ha provocat canvis en els resultats, els quals s'exposen a continuació.

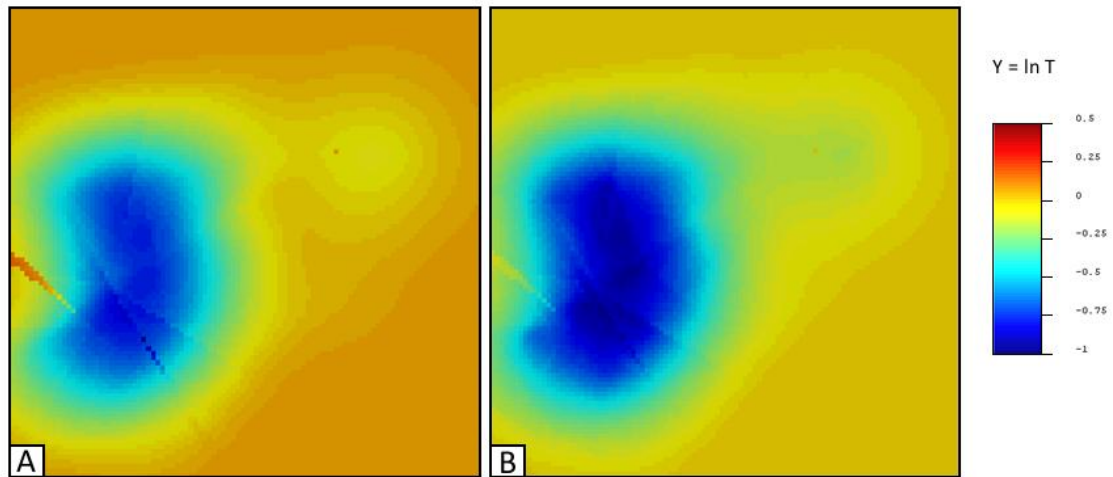


Figura 15: Mapes de Y generats a partir de 8 dades mesurades de τ . A) Resolució = 1, B) Resolució = 0.1

A tots els mapes generats fins el moment, s'ha utilitzat una resolució de 1, i en aquest apartat s'ha optat per augmentar-la per comprovar si les discontinuïtats eren causades per culpa d'una malla massa grollera. Com es pot veure a la figura 15, l'augment de la resolució de la malla provoca que les discontinuïtats es suavitzin, així com també succeeix per a la definició del mapa en general. El mapa 15B té una resolució de 0.1, i per tant el grau de suavitat assolit és destacable, tot i que no soluciona les anomalies, només les difumina i les fa més acord amb la resta del mapa.

Per poder treure una lectura segura d'aquest anàlisi, s'ha repetit el procés però en mapes amb setze dades mesurades, que al tenir més anomalies pot ser més susceptible al canvi de resolució.

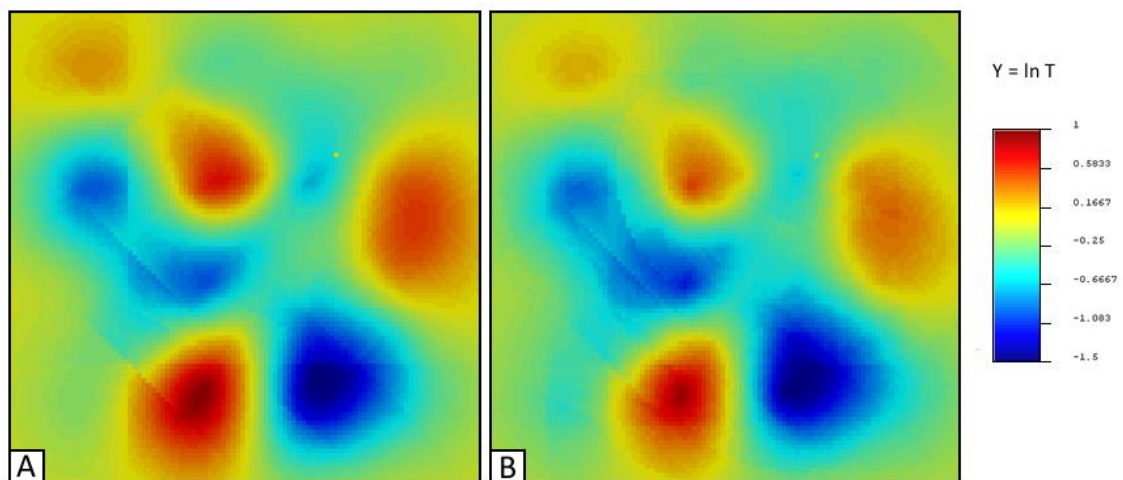


Figura 16: Mapes de Y generats a partir de 16 dades mesurades de τ . A) Resolució = 1, B) Resolució = 0.1

Efectivament com es desprèn de la figura 16, el canvi de resolució té un efecte visible i important en el resultats. Al mapa 16A s'observen discontinuïtats brusques entre zones, que separen valors de transmissivitats molt diferents, mentre que al mapa 16B aquestes discontinuïtats es suavitzen i no hi ha un canvi tant brusc de valors de transmissivitat.

Vist que l'augment de resolució mitiga les anomalies s'ha intentant augmentar-la un ordre més, però el programa dóna errors de càlcul, alertant de la singularitat de certs blocs de les matrius al calcular els pesos.

Així doncs després de l'anàlisi s'ha pogut confirmar que les anomalies es deuen a un problema numèric, i que l'algoritme utilitzat té problemes per a calcular els mapes de transmissivitat amb l'augment de mesures dinàmiques, així com l'augment de la resolució, que podria solucionar el problema.

6. CONCLUSIONS

Amb aquesta nova metodologia s'ha aconseguit estimar i representar mapes de transmissivitat mitjançant, únicament, mesures dinàmiques de temps d'arribada amb prou fidelitat respecte el mapa de referència (figura 1).

També cal tenir en compte que els mapes estimats amb la nova metodologia representen igual o millor, en tots els casos, els mapes de transmissivitats resultants en comparació amb el mapa de referència. És cert que existeixen certs problemes numèrics, però els resultats per a un nombre petit de punts coneguts d'entrada són plenament llegibles i interpretables. A més es creu que refinant l'algoritme del programa es podria resoldre aquest problema, pel que es tracta d'una futura via d'estudi.

Tanmateix també s'ha comprovat que una selecció adequada de la informació d'entrada, pel que fa la localització i distribució, permet l'obtenció de mapes de transmissivitats on es pot fer una lectura aproximada, però veraç, de la tendència de la transmissivitat de cada zona de l'aquífer.

També s'ha vist que la posició del pou no influeix en l'estimació de punts, i que aquesta es fa en línia recta entre el pou i el punt estimat. Es considera que aquest aspecte també podria ser millorat, i que per tant s'hi podria treballar en futurs estudis.

Finalment es pot arribar a la conclusió que la nova metodologia és òptima per estimar camps de transmissivitats, ja que redueix la incertesa globalment i, sobretot al voltant de pou d'extracció, que és la zona que més interès suscita.

7. BIBLIOGRAFIA

1. **Trinchero, Paolo.** *Characterization of Spatial Heterogeneity in Groundwater Applications*. Ph.D. thesis. Technical University of Catalonia : s.n., 2009.
2. **Montero, J.M.; Fernández-Avilés, G.; Mateu, J.; Porcu, E.** Geoestadística espacial y espacio-temporal: vino nuevo, cepas viejas. [autor llibre] AA.VV. *Historia de la Probabilidad y la Estadística (V)*. s.l. : AHEPE, 2011, p. 166-178.
3. **Matheron, G.** *Traité de Géostatistique Appliquée*. París : Editions Technip, 1962. 14.
4. **Institut d'estudis catalans.** DIEC2. *Diccionaria de la llengua catalana. Segona Edició*. [En línia] 14 / juny / 2012. [Data: 12 / abril / 2012.] <http://www.iec.cat/activitats/entrada.asp>.
5. *Connectivity metrics for subsurface flow and transport*. **Renard, Philippe i Allard, Denis**. doi:10.1016/j.advwatres.2011.12.001, Adv Water Resources : s.n., 2011.
6. *Stochastic analysis of spatial variability in subsurface flows-1. Comparisons of one and three-dimensional flows*. **Bakr, AA., et al.** 14(2), 1979, Water Resources Research, p. 263-71.
7. *Spatial variability and uncertainty in groundwater flow parameters*. **Delhomme, J.P.** 15(2), 1979, Water Resources Research, p. 281-90.
8. **Dagan, G.** *Flow and transport in porous formations*. New York : Springer-Verlag, 1989.
9. **Gelhar, L.W.** *Stochastic subsurface hydrology*. Engelwood Cliffs : Prentice-Hall, 1993.
10. **Zhang, D.** *Stochastic methods for flow in porous media*. San Diego : CA: Academic Press, 2002.
11. **Rubi, Y.** *Applied stochastic hydrogeology*. New York : Oxford University Press, 2003.
12. *To be or not to be multi-Gaussian? A reflection on stochastic hydrogeology*. **Gómez-Hernández, J.J. i Wen, X.H.** 21(1), 1998, Adv Water Resources, p. 47-61.
13. *Towards capturing hydrologically significant connectivity in spatial patterns*. **Western, A.W., Blöschl, G. i Grayson, R.B.** 37(1), 2001, Water Resources Research, p. 83-97.
14. *How well do indicator variograms capture the spatial connectivity of soil moisture?* **Western, A.W., Blöschl, G. i Grayson, R.B.** 12(12), 1998, Hydrol Process, p. 1851-68.

15. *Conditional simulation of complex geological structures using multiple-point statistics.* **Strebel, S.** 34(1), 2002, Math Geol, p. 1-21.
16. *Spatial connectivity: from variograms to multiple-point measures.* **Krishnan, S. i Journel, A.G.** 2003, Math Geol, p. 915–25. 35(8).
17. *On the relationship between indicators of geostatistical, flow and transport connectivity.* **Knudby, C. i Carrera, J.** 2005, Adv Water Resour, p. 405–21. 28(4).
18. *The direct sampling method to perform multiple-point geostatistical simulations.* **Mariethoz, G., Renard, P. i Straubhaar, J.** 46, 2010, Water Resources Research.
19. *An improved parallel multiple-point algorithm using a list approach.* **Straubhaar, J., et al.** 43(3), 2011, Math Geosci, p. 305–28.
20. *Fortran programs for calculating connectivity of three-dimensional numerical models and for ranking multiple realizations.* **Deutsch, C.V.** 24(1), 1998, Comput Geosci, p. 69–76.
21. *Multiple point metrics to assess categorical variable models.* **Boisvert, J.B., Pyrcz, M.J. i Deutsch, C.** 2010, Nat Resour Res, p. 165–75. 19(3).
22. *Validation techniques for geological patterns simulations based on variogram and multiple-point statistics.* **De Iaco, S. i Maggio, S.** 2011, Math Geosci, p. 483–500. 43(4).
23. *3D multiple-point statistics simulation using 2D training images.* **Comunian, A., Renard, P. i Straubhaar, J.** Comput Geosci. doi:10.1016/j.cageo.2011.07.009..
24. *Quantitative morphology and network representation of soil pore structure.* **Vogel, H.J. i Roth, K.** 24(3–4), 2001, Adv Water Resour, p. 233–42.
25. *On the use of apparent hydraulic diffusivity as an indicator of connectivity.* **Knudby, C. i Carrera, J.** 329(3–4), 2006, J Hydrol, p. 377–89.
26. *Static characterizations of reservoirs: refining the concepts of connectivity and continuity.* **Hovadik, J.M. i Larue, D.K.** 13(3), 2007, Petrol Geosci, p. 195–211.
27. *Point-to-point connectivity, an abstract concept or a key issue for risk assessment studies?* **Trincheró, P., Sánchez-Vila, X. i Fernández-García, D.** 31(12), 2008, Adv Water Resour, p. 1742–53.

28. *An indirect assessment on the impact of connectivity of conductivity classes upon longitudinal asymptotic macrodispersivity.* **Fiori, A., et al.** 46, 2010, Water Resources Research.
29. *Connectivity of channelized reservoirs: a modelling approach.* **Larue, D.K. i Hovadik, J.** 12(4), 2006, Petrol Geosci, p. 291–308.
30. *A connectivity index for discrete fracture networks.* **Xu, C., et al.** 38(5), 2006, Math Geol, p. 611–34.
31. **King, P.R.** The connectivity and conductivity of overlapping sand bodies. [autor llibre] North Sea Oil and Gas Reservoirs II. London : Graham & Trotman for the Norwegian Institute of Technology, 1990, p. 353–361.
32. **Allard, D. i group, and the HERESIM.** On the connectivity of two random set models: the truncated Gaussian and the Boolean. *Geostatistics Troia '92 vol.1.* s.l. : A. Soares, 1994, p. 467-478.
33. **Nurafza, P.R., King, P.R. i Masih, i M.** Facies connectivity modelling: analysis and field study. *SPE Europe/EAGE annual conference and exhibition.* 2006.
34. **Kitadinis, P.K.** *Introduction to Geostatistics, applications in hydrogeology.* Cambridge : Cambridge, 1997. 0521 58747 6.
35. **Giraldo Henao, Ramon.** *Introducción a la Geoestadística: Teoría y Aplicación.* 2002. Evento: simposio nacional de estadística Ponencia: Libro:, , p. - , v. <, fasc..
36. *On the problem of permissible covariance and variogram models.* **Christakos, G.** 20(2), 1984, Water Resources Research, p. 251-65.
37. **Cressie, N.A.C.** *Statistics for Spatial Data.* Wiley, New York : s.n., 1993.
38. *Pumping tests in heterogeneous aquifers: An analytical study of what can be obtained from their interpretation using jacob's method.* **Sanchez-Vila, X., Meier, P.M. i Carrera, J.** 35(4), Water Resources Research, p. 943-952.
39. **Chapman, Stephen J.** *Fortran 90/95 for scientists and engineers.* s.l. : McGraw-Hill, 2004. 007-123233-8.
40. *El concepto geoestadístico de variable regionalizada.* **Jorge, A. i Rodríguez, G.** 024, Caracas : s.n., 1999, Terra Nueva Etapa, Vol. XV, p. 141-148. 1012-7089.

ANNEX A: CODI PRINCIPAL

En aquest annex s'inclou el codi principal utilitzat per a fer el Kriging, que necessita uns arxius de paràmetres d'entrada i una subrutina que solucioni el problema matricial, els quals s'adjunten als següents annexes.

```

program main
-----
c
c
c               Kriging of a 2-D Rectangular Grid
c               *****
c
c The program is executed with no command line arguments.  The user
c will be prompted for the name of a parameter file.  The parameter
c file is described in the documentation (see the example kb2d.par)
c
c
c
c-----
--

  use cal_elapsed_time

  include 'kb2d.inc'

  print *, '***** '
  print *, ' kb2d version corrected bug '
  print *, '***** '

c
c Read the Parameter File and the Data:
c
  call readparm

c Start clock
  call get_start_time ( ibdt ) ! start clock
c
c Call kb2d to kriging the grid:
c
  call kb2d
c
c Finished:
c
  write(*,9998) VERSION
9998 format(/' KB2D Version: ',f6.3, ' Finished'/)

c Calculate Elapsed Time

  call get_end_time (ibdt)

  stop
  end

subroutine readparm

```

```

C-----
--
C
C               Initialization and Read Parameters
C               *****
C
C The input parameters and data are read in from their files. Some
quick
C error checking is performed and the statistics of all the variables
C being considered are written to standard output.
C
C
C-----
--
    include 'kb2d.inc'
    parameter (MV=20)
    real      var (MV)
    character datafl*132,outfl*132,dbgfl*132,str*132
    logical   testfl
    integer   vrt2

C
C Unit numbers:
C
    lin  = 1
    lout = 2
    ldbg = 3

C
C Note VERSION number:
C
    write(*,9999) VERSION
    9999 format(/' Modified KB2D for Safeguard Zone, Version: ',f6.3/)
C
C Get the name of the parameter file - try the default name if no
input:
C
    write(*,*) 'Which parameter file do you want to use?'
    read (*,'(a40)') str
    if(str(1:1).eq.' ')str='kb2d.par'
,
    inquire(file=str,exist=testfl)
    if(.not.testfl) then
        write(*,*) 'ERROR - the parameter file does not exist,'
        write(*,*) '          check for the file and try again '
        write(*,*)
        if(str(1:20).eq.'kb2d.par' ) then
            write(*,*) '          creating a blank parameter file'
            call makepar
            write(*,*)
        end if
        stop
    endif
    open(lin,file=str,status='OLD')

C
C Find Start of Parameters:
C
1   read(lin,'(a4) ',end=98) str(1:4)
    if(str(1:4).ne.'STAR') go to 1

C
C Read Input Parameters:
C

```



```

read(lin,'(a40)',err=98) datafl
call chknam(datafl,40)
write(*,*) ' data file = ',datafl
pause
C
C  ivar2 is a flaflg that indicates type of variable
C

read(lin,*,err=98)          ixl,iyl,ivrl,ivr2
write(*,*) ' columns for X,Y, VR, VR2 = ',ixl,iyl,ivrl,ivr2

read(lin,*,err=98)          tmin,tmax
write(*,*) ' trimming limits = ',tmin,tmax

read(lin,*,err=98)          idbg
write(*,*) ' debugging level = ',idbg

read(lin,'(a40)',err=98) dbgfl
call chknam(dbgfl,40)
write(*,*) ' debugging file = ',dbgfl

read(lin,'(a40)',err=98) outfl
call chknam(outfl,40)
write(*,*) ' output file = ',outfl

read(lin,*,err=98)          nx,xmn,xsiz
write(*,*) ' nx, xmn, xsiz = ',nx,xmn,xsiz

read(lin,*,err=98)          ny,ymn,ysiz
write(*,*) ' ny, ymn, ysiz = ',ny,ymn,ysiz

!   read(lin,*,err=98)          nxdis,nydis
!   write(*,*) ' discretization = ',nxdis,nydis

read(lin,*,err=98) xwell,ywell
write(*,*) ' well coordinates = ',xwell,ywell

read(lin,*,err=98) resol
write(*,*) ' integral resolution (points/length) = ',resol

nxdis = 1
nydis = 1

read(lin,*,err=98)          ndmin,ndmax
if(ndmin.lt.0) ndmin = 0
write(*,*) ' min max data = ',ndmin,ndmax

read(lin,*,err=98)          radius
write(*,*) ' isotropic radius = ',radius

read(lin,*,err=98)          ktype,skmean
write(*,*) ' ktype,skmea = ',ktype,skmean

read(lin,*,err=98)          nst,c0
write(*,*) ' nst, nugget = ',nst,c0

if(nst.le.0) then
    nst      = 1
    it(1)    = 1
    cc(1)    = 0.0
    ang(1)   = 0.0
    aa(1)    = 0.0

```

```

        anis(1) = 0.0
    else
        do i=1,nst
            read(lin,*,err=98) it(i),cc(i),ang(i),aa(i),a2
            anis(i) = a2 / aa(i)
            write(*,*) ' it,cc,ang,a_max,a_min = ',
                it(i),cc(i),ang(i),aa(i),a2
            if(it(i).eq.4) then
                if(aa(i).lt.0.0) stop ' INVALID power
varioqram'
                if(aa(i).gt.2.0) stop ' INVALID power
varioqram'
            end if
        end do
    end if

    close(lin)
    write(*,*)
    if(nst.gt.MAXNST) stop 'nst is too big - modify .inc file'
    if(ndmax.gt.MAXSAM) stop 'ndmax is too big - modify .inc file'
c
c Check to make sure the data file exists, then either read in the
c data or write an error message and stop:
c
    inquire(file=datafl,exist=testfl)
    if(.not.testfl) then
        write(*,*) 'ERROR data file ',datafl,' does not exist!'
        stop
    endif
c
c The data file exists so open the file and read in the header
c information. Initialize the storage that will be used to summarize
c the data found in the file:
c
    open(lin,file=datafl,status='OLD')
    read(lin,'(a40)',err=99) str
    read(lin,*,err=99) nvari
    av = 0.0
    ss = 0.0
    do i=1,nvari
        read(lin,'(a40)',err=99) str
    end do
c
c Read the data:
c
    nd = 0
7   read(lin,*,end=8,err=99) (var(j),j=1,nvari)
    vrt = var(ivr1)
    if(ivr2.ne.0) then
        vrt2 = int(var(ivr2))
        if(vrt2>2) then
            write(*,*) ' ERROR: variable flag not correct'
            stop
        end if
    else
        vrt2 = 1
    end if
    if(vrt.lt.tmin.or.vrt.ge.tmax) go to 7
    if(vrt2.eq.0) go to 7
    nd = nd + 1
    if(nd.gt.MAXDAT) then

```

```

        write(*,*) ' ERROR: Exceeded available memory for data'
        stop
    end if
    x(nd) = var(ix1)
    y(nd) = var(iy1)
    vr(nd) = vrt
    vr2(nd) = vrt2
c
c         vr2 = 1 indicates variable is Y=lnK
c         vr2 = 2 indicates variable is taup=ln ta -
ln(pi*ri2*b*poro/Qw)
c
    if (vrt2.eq.1) then
        av = av + vrt
        ss = ss + vrt*vrt
    end if
    go to 7
8    close(lin)
c
c Open the output files:
c
    open(lout,file=outfl,status='UNKNOWN')
    write(lout,300)
300 format('KB2D Output',/, '2',/, 'Estimate',/, 'Estimation Variance')
    open(ldbg,file=dbgfl,status='UNKNOWN')
c
c Compute the averages and variances as an error check for the user:
c
    av = av / max(real(nd),1.0)
    ss =(ss / max(real(nd),1.0)) - av * av
c
c Write Some of the Statistics to the screen:
c
    write(*,900) nd,av,sqrt(max(ss,0.0))
900 format(/' There are ',i8,' data with:',/,
+         ' mean value = ',f12.5,/,
+         ' standard deviation = ',f12.5,/)
    return
c
c Error in an Input File Somewhere:
c
98 stop 'ERROR in parameter file!'
99 stop 'ERROR in data file!'
end

```

```

subroutine kb2d
c-----
--
c
c         Ordinary/Simple Kriging of a 2-D Rectangular Grid
c         *****
c
c This subroutine estimates point or block values of one variable by
c ordinary kriging. All of the samples are rescanned for each block
c estimate; this makes the program simple but inefficient. The data
c should NOT contain any missing values. Unestimated points are
c returned as -1.0e21
c

```

```

c
c
c Original:  A.G. Journel
1978
c Revisions: B.E. Buxton
1983
c-----
--

    include 'kb2d.inc'
    real      xdb (MAXDIS), ydb (MAXDIS), xa (MAXSAM), ya (MAXSAM),
+           vra (MAXSAM), dist (MAXSAM)
    real*8    r (MAXSAM+1), rr (MAXSAM+1), s (MAXSAM+1), a (MAXKRG)
    integer   nums (MAXSAM), vra2 (MAXSAM)
    logical   first
    data      first/.true./, PMX/9999.0/
    real      rdist, Ri, xpto, ypto
    integer   nptos, ipto

c
c Echo the input parameters if debugging flag is >2:
c
    if (ldbg.gt.2) then
        write (ldbg,*) 'KB2D Parameters'
        write (ldbg,*)
        write (ldbg,*) 'Variogram Parameters for ',nst,'
structures:'
        write (ldbg,*) '  Nugget effect:           ',c0
        write (ldbg,*) '  Types of variograms:      ',(it(i),i=1,nst)
        write (ldbg,*) '  Contribution cc          ',(cc(i),i=1,nst)
        write (ldbg,*) '  Ranges:                  ',(aa(i),i=1,nst)
        write (ldbg,*) '  Angle for Continuity:    ',(ang(i),i=1,nst)
        write (ldbg,*) '  Anisotropy Factors:
', (anis(i),i=1,nst)
        write (ldbg,*) ' '
        write (ldbg,*) 'Grid for Kriging:'
        write (ldbg,*) '  Number of X and Y Blocks:',nx,ny
        write (ldbg,*) '  Origin of X and Y Blocks:',xmn,ymn
        write (ldbg,*) '  Size   of X and Y Blocks:',xsiz,ysiz
        write (ldbg,*) ' '
c        write (ldbg,*) 'Discretization of blocks: ',nxdis,nydis
        write (ldbg,*) 'Search Radius:           ',radius
        write (ldbg,*) 'Minimum number of samples: ',ndmin
        write (ldbg,*) 'Maximum number of samples: ',ndmax
        write (ldbg,*) ' '
    endif

c
c Echo the input data if debugging flag >1:
c
    if (ldbg.ge.4) then
        do id=1,nd
            write (ldbg,99) id,x(id),y(id),vr(id)
99          format('Data: ',i5,' at ',2f12.3,' value: ',f12.5)
        end do
    endif

c
c Set up the discretization points per block.  Figure out how many
c are needed, the spacing, and fill the xdb and ydb arrays with the
c offsets relative to the block center (this only gets done once):
c
    ndb = nxdis * nydis
    if (ndb.gt.MAXDIS) then
        write (*,*) 'ERROR KB2D: Too many discretization points '

```

```

        write(*,*) '          Increase MAXDIS or lower n[xy]dis'
        stop
    endif
    xdis = xsiz / max(real(nxdis),1.0)
    ydis = ysiz / max(real(nydis),1.0)
    xloc = -0.5*(xsiz+xdis)
    i = 0
    do ix=1,nxdis
        xloc = xloc + xdis
        yloc = -0.5*(ysiz+ydis)
        do iy=1,nydis
            yloc = yloc + ydis
            i = i+1
            xdb(i) = xloc
            ydb(i) = yloc
        end do
    end do

c
c Initialize accumulators:
c
    cbb = 0.0
    rad2 = radius*radius

c
c Calculate Block Covariance. Check for point kriging.
c
    cov = cova2(xdb(1),ydb(1),xdb(1),ydb(1),nst,c0,PMX,cc,
+          aa,it,ang,anis,first)

c
c Keep this value to use for the unbiasedness constraint:
c
    unbiased = cov
    first = .false.

c*****
c*****
c      estimate well column and row and skip this block
c*****
c*****
    icolwell = int(xwell/xsiz)+1
    irowwell = int(ywell/ysiz)+1
c*****
c*****

c
c MAIN LOOP OVER ALL THE BLOCKS IN THE GRID:
c
    nk = 0
    ak = 0.0
    vk = 0.0

    do 4 iy=1,ny

        yloc = ymn + (iy-1)*ysiz

        do 4 ix=1,nx

            xloc = xmn + (ix-1)*xsiz

c
c      skip well cell (Dani)
c
            if(icolwell.eq.ix .and.irowwell.eq.iy) then

```

```

        est  = 0.0
        estv = 0.0
        goto 1
    end if

c
c      calculate variance of arrival time(2)/transmissivity(1)
c
cbb = avgcova2(xloc,yloc,xloc,yloc,nst,c0,PMX,cc,aa,it,
+          ang,anis,first,1,1,xwell,ywell,resol,alpha,ktype,
+          xsiz,ysiz,nx,ny)

    if(idbg.gt.1) then
        write(ldbg,*) ' '
        write(ldbg,*) 'Block Covariance: ',cbb
        write(ldbg,*) ' '
    endif

c
c Find the nearest samples within each octant: First initialize
c the counter arrays:
c
        na = 0
        do isam=1,ndmax
            dist(isam) = 1.0e+20
            nums(isam) = 0
        end do

c
c Scan all the samples (this is inefficient and the user with lots of
c data should move to ktb3d):
c
        do 6 id=1,nd
            dx = x(id) - xloc
            dy = y(id) - yloc
            h2 = dx*dx + dy*dy
            if(h2.gt.rad2) go to 6

c
c Do not consider this sample if there are enough close ones:
c
            if(na>0) then
                if(na.eq.ndmax.and.h2.gt.dist(na)) go to 6
            end if

c
c Consider this sample (it will be added in the correct location):
c
            if(na.lt.ndmax) na = na + 1
            nums(na)          = id
            dist(na)          = h2
            if(na.eq.1) go to 6

c
c Sort samples found thus far in increasing order of distance:
c
            n1 = na-1
            do ii=1,n1
                k=ii
                if(h2.lt.dist(ii)) then
                    jk = 0
                    do jj=k,n1
                        j = n1-jk
                        jk = jk+1
                        j1 = j+1
                        dist(j1) = dist(j)

```

```

                                nums(j1) = nums(j)
                                end do
                                dist(k) = h2
                                nums(k) = id
                                go to 6
                            endif
                        end do
6          continue
c
c Is there enough samples?
c
        if(na.lt.ndmin) then
            if(idbg.ge.2)
                write(ldbg,*) 'Block ',ix,iy, 'not estimated'
                est = UNEST
                estv = UNEST
                go to 1
            endif
c
c Put coordinates and values of neighborhood samples into xa,ya,vra:
c
        do ia=1,na
            jj      = nums(ia)
            xa(ia)  = x(jj)
            ya(ia)  = y(jj)
            vra(ia) = vr(jj)
            vra2(ia) = vr2(jj)
        end do

c Solve the Kriging System with more than one sample:
c
        neq = na + ktype
        nn  = (neq + 1)*neq/2

c
c Set up kriging matrices:
c
        in=0

        do j=1,na
c
c Establish Left Hand Side Covariance Matrix (New):
c
            do i=1,j

                                in = in + 1

                                a(in) =
dble(avgcova2(xa(i),ya(i),xa(j),ya(j),
+                                nst,c0,PMX,cc,aa,it,ang,anis,first,
+                                vra2(i),vra2(j),xwell,ywell,
+                                resol,alpha,ktype,xsiz,ysiz,nx,ny))

                                end do

            xx = xa(j) - xloc
            yy = ya(j) - yloc

c
c Establish Right Hand Side Covariance (New):
c

```

```
        cb = avgcova2(xloc,yloc,xa(j),ya(j),  
nst,c0,PMX,cc,aa,it,ang,anis,first,  
l,vra2(j),xwell,ywell,  
resol,alpha,ktype,xsiz,ysiz,nx,ny)  
  
r(j)   = dble(cb)  
rr(j)  = r(j)  
  
end do  
  
C Set the unbiasedness constraint (sum of weights equal to one):  
C  
    if(ktype.eq.1) then  
        do i=1,na  
            in      = in + 1  
            a(in) = dble(unbias)  
        end do  
        in          = in + 1  
        a(in)       = 0.0  
        r(neq)     = dble(unbias)  
        rr(neq)    = r(neq)  
    end if  
  
C Write out the kriging Matrix if Seriously Debugging:  
C  
    if(idbg.ge.3) then  
        write(ldbg,101) ix,iy  
        is = 1  
        do i=1,neq  
            ie = is + i - 1  
            write(ldbg,102) i,r(i),(a(j)),j=is,ie  
            is = is + i  
        end do  
        format(/,'Kriging Matrices for Node: ',2i4,  
               ' RHS first')  
        format('      r(',i2,',) =',f7.4,' a=  
' ,9(10f7.4))  
  
    endif  
  
C Solve the Kriging System:  
C  
    call ksol(1,neq,1,a,r,s,isng)  
  
C Write a warning if the matrix is singular:  
C  
    if(ising.ne.0) then  
        write(*,*) 'WARNING KB2D: singular matrix'  
        write(*,*) '                      for block',ix,iy  
        est = UNEST  
        estv = UNEST  
        go to 1  
    endif  
  
C Write the kriging weights and data if requested:  
C  
    if(idbg.ge.2) then  
        write(ldbg,*) '  
        write(ldbg,*) 'BLOCK: ',ix,iy  
        write(ldbg,*) '  
        if(ktype.eq.1) write(ldbg,*)
```



```

+
        ' Lagrange multiplier: ',s(neq)*unbias
write(ldbg,*) ' BLOCK EST: x,y,vr,wt '
do i=1,na
write(ldbg,'(4f12.3)') xa(i),ya(i),vra(i),s(i)
end do
endif

c Compute the estimate and the kriging variance:
c
        est1 = 0.0
        est2 = 0.0
        est3 = 0.0
        estv = cbb
        sumw1 = 0.0
        sumw2 = 0.0
        sumw3 = 0.0
        fmeanloc = 0.0
        na2 = 0
        do i=1,na
            if(vra2(i).eq.1) then
                na2=na2+1
                fmeanloc = fmeanloc + vra(i)
                sumw1 = sumw1 + real(s(i))
                est1 = est1 + real(s(i))*vra(i)
            else if (vra2(i).eq.2) then
                sumw2 = sumw2 + real(s(i))
                est2 = est2 + real(s(i))*vra(i)
            else if (vra2(i).eq.3) then
                sumw3 = sumw3 + real(s(i))
                est3 = est3 + real(s(i))*vra(i)
            end if
            estv = estv - real(s(i)*rr(i))
        end do

        if(na2>0) fmeanloc = fmeanloc / float(na2)
        if(ktype.eq.0) est = est1 + est2 + est3 +
                                (1.-sumw1)*skmean
+
        if(ktype.eq.1) est = est1 + est2 + est3 +
                                (1.-sumw1)*fmeanloc
+
        if(ktype.eq.1) estv = estv - real(s(na+1))*unbias

        if(idbg.ge.2) then
            write(ldbg,*) ' est ',est
            write(ldbg,*) ' estv ',estv
            write(ldbg,*) ' '
        endif

c
c Write the result to the output file:
c
1      continue

        write(lout,'(g15.6,1x,g15.6)') est,estv

        if(est.gt.UNEST) then
            nk = nk + 1
            ak = ak + est
            vk = vk + est*est
        end if

c

```

```

c END OF MAIN LOOP OVER ALL THE BLOCKS:
c
4   continue
   if(nk.ge.1) then
       ak = ak / real(nk)
       vk = vk/real(nk) - ak*ak
       write(ldbg,105) nk,ak,vk
       write(*, 105) nk,ak,vk
105  format(/,'Estimated ',i8,' blocks ',/,
           ' average ',f9.4,/, ' variance ',f9.4,/)
       +
   end if
   return
end

+
real function covar2(x1,y1,x2,y2,nst,c0,PMX,cc,aa,it,
                    ang,anis,first)
-----
--
c
c               Covariance Between Two Transmissivity data Points (2-D
Version)
c               *****
c
c This function returns the covariance associated with a variogram
model
c that is specified by a nugget effect and possibly four different
c nested varigoram structures. The anisotropy definition can be
c different for each of the nested structures (spherical, exponential,
c gaussian, or power).
c
c
c
c INPUT VARIABLES:
c
c   x1,y1      Coordinates of first point
c   x2,y2      Coordinates of second point
c   nst        Number of nested structures (max. 4).
c   c0         Nugget constant (isotropic).
c   PMX        Maximum variogram value needed for kriging when
c               using power model. A unique value of PMX is
c               used for all nested structures which use the
c               power model. therefore, PMX should be chosen
c               large enough to account for the largest single
c               structure which uses the power model.
c   cc(nst)    Multiplicative factor of each nested structure.
c   aa(nst)    Parameter "a" of each nested structure.
c   it(nst)    Type of each nested structure:
c               1. spherical model of range a;
c               2. exponential model of parameter a;
c                   i.e. practical range is 3a
c               3. gaussian model of parameter a;
c                   i.e. practical range is a*sqrt(3)
c               4. power model of power a (a must be gt. 0 and
c                   lt. 2). if linear model, a=1,c=slope.
c   ang(nst)   Azimuth angle for the principal direction of
c               continuity (measured clockwise in degrees from
Y)
c   anis(nst)  Anisotropy (radius in minor direction at 90
degrees

```

```

c          from "ang" divided by the principal radius in
c          direction "ang")
c  first          A logical variable which is set to true if the
c                  direction specifications have changed - causes
c                  the rotation matrices to be recomputed.
c
c
c
c OUTPUT VARIABLES: returns "cova2" the covariance obtained from the
c                  variogram model.
c
c
c
c-----
--
parameter(DTOR=3.14159265/180.0, EPSLON=0.0000001)
real      aa(*), cc(*), ang(*), anis(*), rotmat(4,4), maxcov
integer   it(*)
logical   first
save      rotmat, maxcov

c
c The first time around, re-initialize the cosine matrix for the
c variogram structures:
c
  if(first) then
    maxcov = c0
    do is=1, nst
      azimuth      = (90.0-ang(is))*DTOR
      rotmat(1,is) = cos(azimuth)
      rotmat(2,is) = sin(azimuth)
      rotmat(3,is) = -sin(azimuth)
      rotmat(4,is) = cos(azimuth)
      if(it(is).eq.4) then
        maxcov = maxcov + PMX
      else
        maxcov = maxcov + cc(is)
      endif
    end do
  endif

c
c Check for very small distance:
c
  dx = x2-x1
  dy = y2-y1
  if((dx*dx+dy*dy).lt.EPSLON) then
    cova2 = maxcov
    return
  endif

c
c Non-zero distance, loop over all the structures:
c
  cova2 = 0.0
  do is=1, nst

c
c Compute the appropriate structural distance:
c
    dx1 = (dx*rotmat(1,is) + dy*rotmat(2,is))
    dy1 = (dx*rotmat(3,is) + dy*rotmat(4,is))/anis(is)
    h    = sqrt(max((dx1*dx1+dy1*dy1), 0.0))
    if(it(is).eq.1) then

c
c Spherical model:

```

```

c
c      hr = h/aa(is)
c      if(hr.lt.1.0) cova2 = cova2
c      + cc(is)*(1.-hr*(1.5-.5*hr*hr))
c      else if(it(is).eq.2) then
c
c Exponential model:
c
c      cova2 = cova2 +cc(is)*exp(-h/aa(is))
c      else if(it(is).eq. 3) then
c
c Gaussian model:
c
c      hh=-(h*h)/(aa(is)*aa(is))
c      cova2 = cova2 +cc(is)*exp(hh)
c
c      else
c
c Power model:
c
c      cov1 = PMX - cc(is)*(h**aa(is))
c      cova2 = cova2 + cov1
c
c      endif
c
c      end do
c      return
c      end

c      real function avgcova2(x1,y1,x2,y2,nst,c0,PMX,cc,aa,it,
c      +
ang,anis,first,v1,v2,xwell,ywell,resol,alpha,ktype,
c      +
c      xsize,ysize,nx,ny)
c-----
c-----
c
c      DANI: Covariance between different variables (2-D Version)
c      between transmissivity and travel times (connectivity)
c      between two travel time data points
c      *****
c
c This function returns the covariance associated with a variogram
model
c that is specified by a nugget effect and possibly four different
c nested varigoram structures. The anisotropy definition can be
c different for each of the nested structures (spherical, exponential,
c gaussian, or power).
c
c
c
c INPUT VARIABLES:
c
c   x1,y1      Coordinates of first point
c   x2,y2      Coordinates of second point
c   nst        Number of nested structures (max. 4).
c   c0         Nugget constant (isotropic).
c   PMX        Maximum variogram value needed for kriging when
c               using power model. A unique value of PMX is
c               used for all nested structures which use the
c               power model. therefore, PMX should be chosen
c               large enough to account for the largest single
c               structure which uses the power model.
c   cc(nst)    Multiplicative factor of each nested structure.

```

```

c   aa(nst)           Parameter "a" of each nested structure.
c   it(nst)           Type of each nested structure:
c                     1. spherical model of range a;
c                     2. exponential model of parameter a;
c                       i.e. practical range is 3a
c                     3. gaussian model of parameter a;
c                       i.e. practical range is a*sqrt(3)
c                     4. power model of power a (a must be gt. 0 and
c                       lt. 2). if linear model, a=1,c=slope.
c   ang(nst)          Azimuth angle for the principal direction of
c                       continuity (measured clockwise in degrees from
Y)
c   anis(nst)          Anisotropy (radius in minor direction at 90
degrees
c                       from "ang" divided by the principal radius in
c                       direction "ang")
c   first             A logical variable which is set to true if the
c                       direction specifications have changed - causes
c                       the rotation matrices to be recomputed.
c
c   v1,v2             Integer defining type of variables (Y=1,tau=2)
c
c
c
c OUTPUT VARIABLES: returns "cova2" the covariance obtained from the
c                     variogram model.
c
c
c
c-----
--
real      aa(*),cc(*),ang(*),anis(*),rotmat(4,4),maxcov
real      xwell,ywell,resol,xsize,ysize
integer   it(*),v1,v2,flag,nx,ny
logical   first
real      a_linteger

if(v1.eq.1 .and. v2.eq.1) then

    avgcova2 = dble( cova2(x1,y1,x2,y2,nst,c0,PMX,cc,aa,it,
+      ang,anis,first) )

else if(v1.eq.1.and.v2.eq.2) then

c   calculates covariance by means of analytical solutions

    if (nst.eq.1 .and. it(1).eq.1 .and. anis(1).eq.1.) then

        avgcova2=0.
        call sing_points (xwell,ywell,x2,y2,x1,y1,aa(1),
+      Ri,Rj,Rij,rinf,rsup,xinf,yinf,xsup,ysup,flag)
        if (flag.ne.0) return
c
c   *****
c   analytical solution (not functional)
c   *****
        avgcova2 = -a_linteger (rinf,rsup,aa(1),Ri,Rj,Rij)
c   *****
        nptos = resol * (rsup-rinf)
        do ipt=1,nptos
            xpto = xinf + (float(ipt)-0.5)*

```

```

+      (xsup-xinf)/float(nptos)
+      ypto = yinf + (float(ipto)-0.5)*
+              (ysup-yinf)/float(nptos)
+      rdist=sqrt((xpto-xwell)**2+(ypto-ywell)**2)
+
+      if(ktype.eq.0) then
+          avgcova2 = avgcova2 -
+              2.* rdist / (Ri**2) *
+              cova2(x1,y1,xpto,ypto,nst,c0,
+                  PMX,cc,aa,it,ang,anis,first)*
+              (rsup-rinf) /float(nptos)
+      else if(ktype.eq.1) then
+          avgcova2 = avgcova2 +
+              2.* rdist / (Ri**2) *
+              cova2(x1,y1,xpto,ypto,nst,c0,
+                  PMX,cc,aa,it,ang,anis,first)*
+              (rsup-rinf) /float(nptos)
+      end if
+
+      end do
+
+  else
+
+  calculates covariance by means of numerical integration
+
+      avgcova2 = 0.
+      Ri = sqrt((x2-xwell)**2+(y2-ywell)**2)
+      nptos = resol * Ri
+      do ipto=1,nptos
+          xpto = xwell + (float(ipto)-0.5)*
+                      (x2-xwell)/float(nptos)
+          ypto = ywell + (float(ipto)-0.5)*
+                      (y2-ywell)/float(nptos)
+          rdist=sqrt((xpto-xwell)**2+(ypto-ywell)**2)
+
+          if(ktype.eq.0) then
+              avgcova2 = avgcova2 -
+                  2.* rdist / (Ri**2) *
+                  cova2(x1,y1,xpto,ypto,nst,c0,
+                      PMX,cc,aa,it,ang,anis,first)*
+                  Ri/float(nptos)
+          else if(ktype.eq.1) then
+              avgcova2 = avgcova2 +
+                  2.* rdist / (Ri**2) *
+                  cova2(x1,y1,xpto,ypto,nst,c0,
+                      PMX,cc,aa,it,ang,anis,first)*
+                  Ri/float(nptos)
+          end if
+
+      end do
+
+  end if
+
+  else if(v1.eq.2.and.v2.eq.1) then
+
+  calculates covariance by means of analytical solutions
+
+      if (nst.eq.1 .and. it(1).eq.1 .and. anis(1).eq.1.) then
+
+          avgcova2=0.
+
+          call sing points (xwell,ywell,x1,y1,x2,y2,aa(1),

```

```

C                                     Ri,Rj,Rij,rinf,rsup,xinf,yinf,xsup,ysup,flag)

        if (flag.ne.0) return
C *****
analytical solution
C     avgcova2 = -a_linteger (rinf,rsup,aa(1),Ri,Rj,Rij)
C     *****
        nptos = resol * (rsup-rinf)
        do ipto=1,nptos
            xpto = xinf + (float(ipto)-0.5)*
+                               (xsup-xinf)/float(nptos)
            ypto = yinf + (float(ipto)-0.5)*
+                               (ysup-yinf)/float(nptos)
            rdist=sqrt((xpto-xwell)**2+(ypto-ywell)**2)

            if(ktype.eq.0) then
+               avgcova2 = avgcova2 -
+               2.* rdist / (Ri**2) *
+               (cova2(x2,y2,xpto,ypto,nst,c0,
+               PMX,cc,aa,it,ang,anis,first))*
+               (rsup-rinf) /float(nptos)
            else if(ktype.eq.1) then
+               avgcova2 = avgcova2 +
+               2.* rdist / (Ri**2) *
+               (cova2(x2,y2,xpto,ypto,nst,c0,
+               PMX,cc,aa,it,ang,anis,first))*
+               (rsup-rinf) /float(nptos)
            end if

        end do
    else
C     calculates covariance by means of numerical integration

        avgcova2=0.
        Ri = sqrt((x1-xwell)**2+(y1-ywell)**2)
        nptos = resol * Ri
        do ipto=1,nptos
            xpto = xwell + (float(ipto)-0.5)*
+                               (x1-xwell)/float(nptos)
            ypto = ywell + (float(ipto)-0.5)*
+                               (y1-ywell)/float(nptos)
            rdist=sqrt((xpto-xwell)**2+(ypto-ywell)**2)

            if(ktype.eq.0) then
+               avgcova2 = avgcova2 -
+               2.* rdist / (Ri**2) *
+               cova2(x2,y2,xpto,ypto,nst,c0,
+               PMX,cc,aa,it,ang,anis,first)*
+               Ri/float(nptos)
            else if(ktype.eq.1) then
+               avgcova2 = avgcova2 +
+               2.* rdist / (Ri**2) *
+               cova2(x2,y2,xpto,ypto,nst,c0,
+               PMX,cc,aa,it,ang,anis,first)*
+               Ri/float(nptos)
            end if

        end do

```

```

        end if

elseif (v1.eq.2.and.v2.eq.2) then

c      calculates covariance by means of analytical solutions

        if (nst.eq.1 .and. it(1).eq.1 .and. anis(1).eq.1.) then

            avgcova2=0.
            Rxi=sqrt((x1-xwell)**2+(y1-ywell)**2)
            Rxj=sqrt((x2-xwell)**2+(y2-ywell)**2)
            niptos = resol * Rxi
            do ipto=1,niptos
                xptol = xwell + (float(ipto)-0.5)*
+                               (x1-xwell)/float(niptos)
+                               (y1-ywell)/float(niptos)
                yptol = ywell + (float(ipto)-0.5)*
+                               (y1-ywell)/float(niptos)
                rdist1=sqrt((xptol-xwell)**2+
+                               (yptol-ywell)**2)
                call sing_points (xwell,ywell,x2,y2,xptol,yptol,
+                               aa(1),Ri,Rj,Rij,rinf,rsup,xinf,
+                               yinf,xsup,ysup,flag)
                if (flag.ne.0) return
                njptos = resol * (rsup-rinf)
                do jpto=1,njptos
                    xpto2 = xinf + (float(jpto)-0.5)*
+                               (x2-xinf)/float(njptos)
                    ypto2 = yinf + (float(jpto)-0.5)*
+                               (y2-yinf)/float(njptos)
                    rdist2=sqrt((xpto2-xwell)**2+
+                               (ypto2-ywell)**2)
                    avgcova2 = avgcova2 +
+                               4.*rdist1*rdist2/(Rxi**2)/(Rxj**2)*
+                               cova2(xptol,yptol,xpto2,ypto2,nst,c0,
+                               PMX,cc,aa,it,ang,anis,first)*
+                               (Rxi/float(niptos))*
+                               ((rsup-rinf)/float(njptos))

                end do
            end do
        else

c      calculates covariance by means of numerical integration

            Rxi=sqrt((x1-xwell)**2+(y1-ywell)**2)
            Rxj=sqrt((x2-xwell)**2+(y2-ywell)**2)
            niptos = resol * Rxi
            njptos = resol * Rxj
            do ipto=1,niptos
                xptol = xwell + (float(ipto)-0.5)*
+                               (x1-xwell)/float(niptos)
+                               (y1-ywell)/float(niptos)
                yptol = ywell + (float(ipto)-0.5)*
+                               (y1-ywell)/float(niptos)
                rdist1=sqrt((xptol-xwell)**2+
+                               (yptol-ywell)**2)
            do jpto=1,njptos
                xpto2 = xwell + (float(jpto)-0.5)*

```



```

+                                     (x2-xwell)/float(njptos)
ypto2 = ywell + (float(jpto)-0.5)*
+                                     (y2-ywell)/float(njptos)
+                                     rdist2=sqrt((xpto2-xwell)**2+
+                                     (ypto2-ywell)**2)
+                                     avgcova2 = avgcova2 +
+                                     4.*rdist1*rdist2/(Rxi**2)/(Rxj**2)*
+                                     cova2(xpto1,ypto1,xpto2,ypto2,nst,c0,
+                                     PMX,cc,aa,it,ang,anis,first)*
+                                     (Rxi/float(niptos))*(Rxj/float(njptos))

                                end do
                        end do

                end if

        end if

        return
    end

    real function avgcovaYW(x1,y1,x2,y2,nst,c0,PMX,cc,aa,it,
+    ang,anis,first,xwell,ywell,xsize,ysize,nx,ny)
c-----
c--
c
c      DANI: Covariance between different variables (2-D Version)
c      *****
c
c This function returns the covariance associated with a variogram
model
c that is specified by a nugget effect and possibly four different
c nested varigoram structures. The anisotropy definition can be
c different for each of the nested structures (spherical, exponential,
c gaussian, or power).
c
c
c
c INPUT VARIABLES:
c
c   x1,y1          Coordinates of first point
c   x2,y2          Coordinates of second point
c   nst            Number of nested structures (max. 4).
c   c0             Nugget constant (isotropic).
c   PMX            Maximum variogram value needed for kriging when
c                  using power model. A unique value of PMX is
c                  used for all nested structures which use the
c                  power model. therefore, PMX should be chosen
c                  large enough to account for the largest single
c                  structure which uses the power model.
c   cc(nst)        Multiplicative factor of each nested structure.
c   aa(nst)        Parameter "a" of each nested structure.
c   it(nst)        Type of each nested structure:
c                  1. spherical model of range a;
c                  2. exponential model of parameter a;
c                     i.e. practical range is 3a
c                  3. gaussian model of parameter a;
c                     i.e. practical range is a*sqrt(3)
c                  4. power model of power a (a must be gt. 0 and
c                     lt. 2). if linear model, a=1,c=slope.
c   ang(nst)       Azimuth angle for the principal direction of

```

```

c          continuity (measured clockwise in degrees from
Y)
c  anis(nst)          Anisotropy (radius in minor direction at 90
degrees
c                      from "ang" divided by the principal radius in
c                      direction "ang")
c  first              A logical variable which is set to true if the
c                      direction specifications have changed - causes
c                      the rotation matrices to be recomputed.
c
c  v1,v2              Integer defining type of variables (Y=1,tau=2)
c
c
c
c
c OUTPUT VARIABLES: returns "cova2" the covariance obtained from the
c                      variogram model.
c
c
c
c-----
--
c  SO FAR ONLY ISOTROPIC COVARIANCE FUNCTIONS
c
  implicit none
  real      aa(*),cc(*),ang(*),anis(*)
  real      xwell,ywell,xsize,ysize,x1,x2,y1,y2,PMX,c0
  integer   it(*),nst
  logical   first
  integer   i,j,imin,imax,jmin,jmax,is,nx,ny
  real      amax,sum,xptol,yptol
  real      U_weight,cova2
  real      Ri,Rj,Rij,rinf,rsup,xinf,yinf,xsup,ysup
  integer   flag

  sum=0.
  do is=1,nst
    sum=sum + anis(is)
  end do
  sum = sum/float(nst)

  if ( sum.eq.1.) then

    amax=aa(1)
    do is=1,nst
      if (aa(is).gt.amax) then
        amax = aa(is)
      end if
    end do

    imin = max(int((x1-amax)/xsize)+1,0)
    imax = min(int((x1+amax)/xsize)+1,nx)
    jmin = max(int((y1-amax)/ysize)+1,0)
    jmax = min(int((y1+amax)/ysize)+1,ny)

    avgcovaYW=0.

    amax=0.5*amax

    call sing_points (xwell,ywell,x2,y2,x1,y1,amax,
c                      Ri,Rj,Rij,rinf,rsup,xinf,yinf,xsup,ysup,flag)

```

```

    if (flag.ne.0) return

    do i=imin,imax
      do j=jmin,jmax

        xptol = (float(i)-0.5) * xsize
        yptol = (float(j)-0.5) * ysize
        avgcovaYW = avgcovaYW +
+           U_weight (xwell,ywell,x2,y2,xptol,yptol) *
+           cova2(x1,y1,xptol,yptol,nst,c0,
+           PMX,cc,aa,it,ang,anis,first) * xsize * ysize
      end do
    end do
    avgcovaYW = -avgcovaYW/3.1415926535
  else
    stop ' **** ANISOTROPY IS NOT AVAILABLE *** '
  end if

  return
end

C*****
C*****
C      U FUNCTION:
C
C This function calculates the value of the weight function given the
C position of the well, the
C observation point (xobs,yobs), the perturbed point (xpert,ypert)
C*****
C*****
      real function U_weight (xwell,ywell,xobs,yobs,xpert,ypert)
      implicit none
      real xwell,ywell,xpert,ypert,xobs,yobs
      real erre,elle,erreprima,cosfi
C      erre is the distance between the well and the perturbed point
C      elle is the distance between the well and the observation point
C      (xobs,yobs)
C      erreprima is the distance between the perturbed point and the
C      observation point (xobs,yobs)
      erre=sqrt((xwell-xpert)**2+(ywell-ypert)**2)
      elle=sqrt((xwell-xobs)**2+(ywell-yobs)**2)

      if (erre.eq.0. .or. elle.eq.0.) then
        U_weight = 0.
      else
        erreprima=sqrt((xobs-xpert)**2+(yobs-ypert)**2)
        cosfi=(erre**2+elle**2-erreprima**2)/(2*erre*elle)
        U_weight=((erre-elle*cosfi)/((erreprima**2)))/(erre**2)
      endif

      return
    end

C*****
C*****

      subroutine makepar
C-----
C
C      Write a Parameter File
C      *****
C

```

```

C
C-----
--
    lun = 99
    open(lun,file='kb2d.par',status='UNKNOWN')
    write(lun,10)
10    format('                                Parameters for KB2D',/,
+          '                                *****',/,/,
+          'START OF PARAMETERS:')

    write(lun,11)
11    format(' ../data/cluster.dat           ',
+          '-file with data')
    write(lun,12)
12    format('1    2    4    5                ',
+          '-    columns for X, Y, variable, vartype')
    write(lun,13)
13    format('-1.0e21    1.0e21                ',
+          '-    trimming limits')
    write(lun,14)
14    format('0                                ',
+          '-debugging level: 0,1,2,3')
    write(lun,15)
15    format('kb2d.dbg                        ',
+          '-file for debugging output')
    write(lun,16)
16    format('kb2d.out                        ',
+          '-file for kriged output')
    write(lun,17)
17    format('100    0.5    1.0                    ',
+          '-nx,xmn,xsiz')
    write(lun,18)
18    format('100    0.5    1.0                    ',
+          '-ny,ymn,ysiz')
    write(lun,19)
19    format('69.5    69.5                        ',
+          '- xwell, ywell')
    write(lun,191)
191    format('1.0                                ',
+          '- resolution (points per unit length ')
    write(lun,20)
20    format('50    100                            ',
+          '-min and max data for kriging')
    write(lun,21)
21    format('100.0                                ',
+          '-maximum search radius')
    write(lun,22)

22    format('1    1.45                        ',
+          '-0=SK, 1=OK, (mean if SK)')
    write(lun,23)
23    format('1    0.0                            ',
+          '-nst, nugget effect')
    write(lun,24)
24    format('1    1.0    0.0    30.0    30.0        ',
+          '-it, c, azm, a_max, a_min')

    close(lun)
    return
end

```

ANNEX B: ARXIUS AMB PARÀMETRES D'ENTRADA

Arxius amb paràmetres d'entrades pel programa principal:

Annex B1

```

Parameters for KB2DTIME
*****

START OF PARAMETERS:
Extract_4t.out          \file with data
1  2  4  5              \columns for X, Y, variable,
vartype
-1.0e21  1.0e21         \trimming limits
0              \debugging level: 0,1,2,3
kb2d_July_a.dbg         \file for debugging output
kb2d_July_a.out         \file for kriged output
100  0.5  1.0          \nx,xmn,xsiz
100  0.5  1.0          \ny,ymn,ysiz
10  15                \xwell,ywell
1              \resolution (points per unit length)
1  100            \min and max data for kriging
100            \maximum search radius
0  0              \0=SK, 1=OK, (mean if SK)
1  0.0           \nst, nugget effect
1  1.0  0.0  30.0  30.0 \it, c, azm, a_max, a_min

```

Annex B2

```

c-----
--
c
c
c              2-D Kriging Program
c              *****
c
c The following Parameters control static dimensioning within okb2d:
c
c  MAXX      maximum nodes in X
c  MAXY      maximum nodes in Y
c  MAXDAT    maximum number of data points
c  MAXSAM    maximum number of data points to use in one kriging
system
c  MAXDIS    maximum number of discretization points per block
c  MAXNST    maximum number of nested structures
c
c-----
--
c
c User Adjustable:
c
c      parameter(MAXDAT =10000,
+          MAXSAM = 120,
+          MAXDIS = 64,
+          MAXNST = 4)
c

```

```

c Fixed
c
      parameter (MAXKD=MAXSAM+1,MAXKRG=MAXKD*MAXKD,UNEST=-999.,
+              EPSLON=0.0000001,VERSION=2.000)
c
c Variable Declaration:
c
      real      x (MAXDAT), y (MAXDAT), vr (MAXDAT), aa (MAXNST), cc (MAXNST),
+              ang (MAXNST), anis (MAXNST)
      integer   it (MAXNST), vr2 (MAXDAT), ibdt (8)

      real      xwell, ywell, resol
c
c Common Blocks:
c
      common /datagv/ nd, x, y, vr, vr2

      common /parmfk/ nx, ny, xmn, ymn, xsiz, ysiz, nxdis, nydis, ndmin,
+                  ndmax, radius, nst, c0, it, aa, cc, ang, anis, ktype,
+                  skmean, idbg, lout, ldbg

      common /connectivity/ xwell, ywell, resol, alpha

```

ANNEX C: SUBROUTINES

Subrutina per resoldre el problema matricial, és a dir per solucionar el sistema lineal d'equacions.

```

subroutine ksol(nright,neq,nsb,a,r,s,ising)
c-----
c
c
c          Solution of a System of Linear Equations
c          *****
c
c
c
c INPUT VARIABLES:
c
c   nright,nsb      number of columns in right hand side matrix.
c                   for KB2D: nright=1, nsb=1
c   neq             number of equations
c   a()             upper triangular left hand side matrix (stored
c                   columnwise)
c   r()             right hand side matrix (stored columnwise)
c                   for kb2d, one column per variable
c
c
c
c OUTPUT VARIABLES:
c
c   s()             solution array, same dimension as  r  above.
c   ising           singularity indicator
c                   0,  no singularity problem
c                   -1, neq .le. 1
c                   k,  a null pivot appeared at the kth iteration
c
c
c
c PROGRAM NOTES:
c
c   1. Requires the upper triangular left hand side matrix.
c   2. Pivots are on the diagonal.
c   3. Does not search for max. element for pivot.
c   4. Several right hand side matrices possible.
c   5. USE for ok and sk only, NOT for UK.
c
c-----
c
c
c   implicit real*8 (a-h,o-z)
c   real*8    a(*),r(*),s(*)
c
c If there is only one equation then set ising and return:
c
c   if(neq.le.1) then
c       ising = -1
c       return
c   endif
c
c Initialize:

```

```

c
    tol    = 0.1e-06
    ising  = 0
    nn     = neq*(neq+1)/2
    nm     = nsb*neq
    m1     = neq-1
    kk     = 0

c
c Start triangulation:
c
    do k=1,m1
        kk=kk+k
        ak=a(kk)
        if(abs(ak).lt.tol) then
            ising=k
            return
        endif
        km1=k-1
        do iv=1,nright
            nm1=nm*(iv-1)
            ii=kk+nn*(iv-1)
            piv=1./a(ii)
            lp=0
            do i=k,m1
                ll=ii
                ii=ii+i
                ap=a(ii)*piv
                lp=lp+1
                ij=ii-km1
                do j=i,m1
                    ij=ij+j
                    ll=ll+j
                    a(ij)=a(ij)-ap*a(ll)
                end do
                do llb=k,nm,neq
                    in=llb+lp+nm1
                    ll1=llb+nm1
                    r(in)=r(in)-ap*r(ll1)
                end do
            end do
        end do
    end do

c
c Error checking - singular matrix:
c
    ijm=ij-nn*(nright-1)
    if(abs(a(ijm)).lt.tol) then
        ising=neq
        return
    endif

c
c Finished triangulation, start solving back:
c
    do iv=1,nright
        nm1=nm*(iv-1)
        ij=ijm+nn*(iv-1)
        piv=1./a(ij)
        do llb=neq,nm,neq
            ll1=llb+nm1
            s(ll1)=r(ll1)*piv
        end do
    end do

```



```

i=neq
kk=ij
do ii=1,m1
    kk=kk-i
    piv=1./a(kk)
    i=i-1
    do llb=i,nm,neq
        ll1=llb+nm1
        in=ll1
        ap=r(in)
        ij=kk
        do j=i,m1
            ij=ij+j
            in=in+1
            ap=ap-a(ij)*s(in)
        end do
        s(ll1)=ap*piv
    end do
end do
end do
c
c Finished solving back, return:
c
return
end

```


ANNEX D: SCRIPTS

Scripts utilitzats per extreure les dades conegudes dels camps reals de transmissivitats i temps d'arribada.

Annex D1

```

program nyi

! Proposit:
! Extreure un n° determinat de dades d'un fitxer de dades major
  implicit none

! Aquest programa llegeix n punts des d'un des dun arxiu de dades i
els guarda en 4 arreglos x, y, z, T, k

      CHARACTER (len=20) :: filename, filename2
      CHARACTER (len=20) :: str
      INTEGER :: nvari=0
      INTEGER :: status, i
      INTEGER :: n=0

      REAL :: x, y, z, Field, Type
      Real, dimension (100) :: ax, ay, az, af, at

      WRITE (*,*) 'Please enter input file name'
      READ (*,*) filename
      WRITE (*,1000) filename
      1000 FORMAT (' ', 'The input file name is: ', A)

! Obrir l'arxiu de dades

      OPEN (UNIT=3, FILE=filename, STATUS='OLD', ACTION='READ',
      IOSTAT=status)

!Header Information

      READ (3, '(a40)', IOSTAT=status ) str
      READ (3, *, IOSTAT=status)          nvari
      DO i=1, nvari
          READ(3, '(a40)', IOSTAT=status) str

      END DO

! Extracció de les dades de la mostra total

5 WRITE (*,*) 'How many points do you want to extract, 4, 5, 6 or 8?
(it will be n x n points)'
READ (*,*) n
WRITE (*,1010) n, n
1010 FORMAT (' ', ' You want the following number of points: ', I3, '
x', I3 )

if (n>10) then

print *, 'There are not points enough in the file to extract n x n
points'

```

```

go to 5

end if

WRITE (*,*) 'What is the name of the data file?'
READ (*,*) filename2
WRITE (*,1060) filename2
1060 FORMAT (' ', 'The name of the output file is: ', A)

!Read the Data:

OPEN (UNIT=4, FILE=filename2, STATUS='NEW', ACTION='WRITE',
IOSTAT=status)
WRITE(4,*) 'Randomly extracted data'
WRITE(4,*) '5'
WRITE(4,*) 'x'
WRITE(4,*) 'y'
WRITE(4,*) 'z'
WRITE(4,*) 'Field'
WRITE(4,*) 'Type'

if (n==4) then
  1 do i=1, 100
    READ(3,*,IOSTAT=status) ax(i), ay(i), az(i), af(i), at(i)

    x=ax(i)
    y=ay(i)
    z=az(i)
    field=af(i)
    type=at(i)

    if
      ((i==1).or.(i==4).or.(i==7).or.(i==10).or.(i==31).or.(i==34).or.(i==37)
      ).or.(i==40).or.(i==61).or.(i==64).or.(i==67).or.(i==70).or.(i==91).or
      .(i==94).or.(i==97).or.(i==100)) then

        WRITE(4,1020) x, y, z, Field, Type
        1020 FORMAT ( F10.4, F10.4, F10.4, F10.4, F10.4)

      end if
    end do
    pause
  else if (n==5) then
    2 do i=1, 100
      READ(3,*,IOSTAT=status) ax(i), ay(i), az(i), af(i), at(i)

      x=ax(i)
      y=ay(i)
      z=az(i)
      field=af(i)
      type=at(i)

      if
        ((i==1).or.(i==3).or.(i==5).or.(i==7).or.(i==9).or.(i==21).or.(i==23).
        or.(i==25).or.(i==27).or.(i==29).or.(i==41).or.(i==43).or.(i==45).or.(
        i==47).or.(i==49).or.(i==61).or.(i==63).or.(i==65).or.(i==67).or.(i==6
        9).or.(i==81).or.(i==83).or.(i==85).or.(i==87).or.(i==89)) then

```

```

OPEN (UNIT=4, FILE=filename2, STATUS='NEW',
ACTION='WRITE', IOSTAT=status)

WRITE(4,1030) x, y, z, Field, Type
1030 FORMAT (' ', F10.4, F10.4, F10.4, F10.4, F10.4)

end if
end do
pause

else if (n==6) then
3 do i=1, 100
READ(3,*,IOSTAT=status) ax(i), ay(i), az(i), af(i), at(i)

x=ax(i)
y=ay(i)
z=az(i)
field=af(i)
type=at(i)

if
((i==1).or.(i==3).or.(i==5).or.(i==6).or.(i==8).or.(i==10).or.(i==21).
or.(i==23).or.(i==25).or.(i==26).or.(i==28).or.(i==30).or.(i==41).or.(
i==43).or.(i==45).or.(i==46).or.(i==48).or.(i==50).or.(i==51).or.(i==5
3).or.(i==55).or.(i==56).or.(i==58).or.(i==60).or.(i==71).or.(i==73).o
r.(i==75).or.(i==76).or.(i==78).or.(i==80).or.(i==91).or.(i==93).or.(i
==95).or.(i==96).or.(i==98).or.(i==100)) then

OPEN (UNIT=4, FILE=filename2, STATUS='NEW',
ACTION='WRITE', IOSTAT=status)

WRITE(4,1040) x, y, z, Field, Type
1040 FORMAT (' ', F10.4, F10.4, F10.4, F10.4, F10.4)

end if
end do
pause

else if (n==8) then
4 do i=1, 100
READ(3,*,IOSTAT=status) ax(i), ay(i), az(i), af(i), at(i)

x=ax(i)
y=ay(i)
z=az(i)
field=af(i)
type=at(i)

if
((i==1).or.(i==3).or.(i==4).or.(i==5).or.(i==6).or.(i==7).or.(i==8).or
.(i==10).or.(i==21).or.(i==23).or.(i==24).or.(i==25).or.(i==26).or.(i
==27).or.(i==28).or.(i==30).or.(i==31).or.(i==33).or.(i==34).or.(i==35)
.or.(i==36).or.(i==37).or.(i==38).or.(i==40).or.(i==41).or.(i==43).or.
(i==44).or.(i==45).or.(i==46).or.(i==47).or.(i==48).or.(i==50).or.(i==
51).or.(i==53).or.(i==54).or.(i==55).or.(i==56).or.(i==57).or.(i==58).
or.(i==60).or.(i==61).or.(i==63).or.(i==64).or.(i==65).or.(i==66).or.(
i==67).or.(i==68).or.(i==70).or.(i==71).or.(i==73).or.(i==74).or.(i==7
5).or.(i==76).or.(i==77).or.(i==78).or.(i==80).or.(i==91).or.(i==93).o
r.(i==94).or.(i==95).or.(i==96).or.(i==97).or.(i==98).or.(i==100))
then

```

```

        OPEN (UNIT=4, FILE=filename2, STATUS='NEW',
ACTION='WRITE', IOSTAT=status)

        WRITE(4,1050) x, y, z, Field, Type
1050 FORMAT (' ', F10.4, F10.4, F10.4, F10.4, F10.4)

        end if
    end do
    pause
end if

CLOSE (UNIT=3)

END PROGRAM nyi

```

Annex D2

```

! RandomExtraction.f90
!
! FUNCTIONS:
!   RandomExtraction      - Entry point of console application.
!
! *****
!
! PROGRAM: RandomExtraction
!
! PURPOSE:  Entry point for the console application.
!
! *****

program RandomExtraction

implicit none

character(len=100)    :: fname,name
integer              ::
nx,ny,nz,next,var,next_x,next_y,int_x,int_y,i11,j11,iwell,jwell
integer              :: i,j,k,iseed,ivar,nvar,iter,iext
real*8, allocatable  :: val(:)
integer, allocatable  :: u(:)
real*8               ::
xcoord,ycoord,zcoord,dx,dy,dz,xwell,ywell
real*8, external     :: rand
logical              :: random

print *, ' Tell me file with GSLIB data ?'
read(*,*) fname

open(10,file=fname)
open(20,file='Extract.out')

print *, ' Tell me nx, ny, nz ?'
read(*,*) nx,ny,nz

```

```

    print *, ' Tell me variable to use ?'
    read(*,*) var

print *, ' Tell me dx, dy, dz ?'
read(*,*) dx,dy,dz

print *, ' Tell me number of extractions ?'
read(*,*) next

print *, ' Tell me if you want RANDOM extraction (T/F) ?'
read(*,*) random

xwell = 49.5
ywell = 49.5

iwell = int(xwell/dx)+1
jwell = int(ywell/dy)+1

! random numbers

if (Random) then

    allocate (u(next))

    iseed=12345

    do i=1,next
        u(i)=int(rand(iseed)*nx*ny*nz)
    end do

else if (.not.random) then

    next_x = int(sqrt(float(next*nx) / float(ny) ))
    next_y = int(sqrt(float(next*ny) / float(nx) ))

    allocate(u(4*next))

    int_x = int(nx/next_x + 0.5)
    int_y = int(ny/next_y + 0.5)

    i11 = iwell + int_x/2
    j11 = jwell + int_y/2

    next=0
    do i=i11,nx,int_x
        do j=j11,ny,int_y
            next=next+1
            u(next) = nx*(j-1) + i
        end do
    end do

    i11 = iwell - int_x/2
    j11 = jwell + int_y/2

    do i=i11,1,-int_x
        do j=j11,ny,int_y
            next=next+1
            u(next) = nx*(j-1) + i
        end do
    end do

```

```

        end do
    end do

    i11 = iwell - int_x/2
    j11 = jwell - int_y/2

    do i=i11,1,-int_x
        do j=j11,1,-int_y
            next=next+1
            u(next) = nx*(j-1) + i
        end do
    end do

    i11 = iwell + int_x/2
    j11 = jwell - int_y/2

    do i=i11,nx,int_x
        do j=j11,1,-int_y
            next=next+1
            u(next) = nx*(j-1) + i
        end do
    end do

end if

print *, ' Number of extractions = ', next

! write head output file

write(20,*) ' Randomly extracted data'
write(20,*) ' 5'

write(20,*) ' x'
write(20,*) ' y'
write(20,*) ' z'
write(20,*) ' Field'
write(20,*) ' Type'

! read gslib file and write

read(10,*)
read(10,*) nvar

allocate (val(nvar))

do ivar=1,nvar
    read(10,*) name
end do

iter=0
do k=1,nz
    do j=1,ny
        do i=1,nx
            iter=iter+1
            read(10,*) (val(ivar),ivar=1,nvar)
            do iext=1,next
                if (iter == u(iext)) then
                    xcoord = (dfloat(i)-0.5d0)*dx
                    ycoord = (dfloat(j)-0.5d0)*dy
                    zcoord = (dfloat(k)-0.5d0)*dz

```